

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»**
(МИ ВлГУ)

Факультет Информационных технологий и радиоэлектроники

Кафедра Программной инженерии

УТВЕРЖДАЮ

Зав. кафедрой

(подпись) Я.Ю. Кульков

« ____ » _____ 2025 г

БАКАЛАВРСКАЯ РАБОТА

Тема Разработка веб-приложения для онлайн-заказа и доставки

продуктов питания

МИВУ.09.03.04-17.000 ВКР

Руководитель

Холкина Н. Е.

(фамилия, инициалы)

(подпись)

(дата)

Студент ПИн –121

(группа)

Скотников Е. С.

(фамилия, инициалы)

(подпись)

(дата)

Муром 2025 г.

Темой выпускной квалификационной работы является разработка веб-приложения для онлайн-заказа и доставки продуктов питания. В данной работе выполняется создание архитектуры системы, а также её реализация. При создании приложения использовались среды разработки IntelliJ IDEA, Android Studio и языки программирования Java, Kotlin и Javascript. В ходе работы были построены, диаграммы вариантов использования, описаны руководства пользователя, администратора, курьера и программиста.

The topic of the final qualification work is the development of a web application for online ordering and delivery of food products. This work involves the creation of the system architecture and its implementation. The application was created using the IntelliJ IDEA and Android Studio development environments and the Java, Kotlin, and Javascript programming languages. During the course of the work, use case diagrams were constructed and user, administrator, courier, and programmer manuals were described.

Содержание

Введение.....	5
1 Анализ технического задания	6
1.1 Описание предметной области.....	6
1.2 Особенности законодательного регулирования.....	7
1.3 Обзор аналогов	8
1.4 Анализ требований к разрабатываемой системе	10
2. Проектирование и разработка системы.....	13
2.1 Моделирование предметной области	13
2.2 Бизнес–требования.....	14
2.3 Моделирование данных	15
2.4 Обоснование выбора СУБД.....	17
2.5 Стек технологий	18
2.6 Физическая модель базы данных	26
3 Описание системы.....	28
3.1 Руководство программиста.....	28
3.2 Руководство пользователя	37
3.3 Руководство администратора	46
3.3 Руководство курьера	50
4 Тестирование системы	52
4.1 Тестирование клиентской части	53
4.2 Тестирование административной части	54
4.3 Тестирование мобильного приложения для курьеров	56
Заключение.....	57
Список литературы.....	58
Приложение А.....	59
Приложение Б	65
Приложение В.....	67

					МИВУ.09.03.04-17.000 ВКР			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Скотников Е. С.			Разработка веб-приложения для онлайн-заказа и доставки продуктов питания Пояснительная записка	Лит.	Лист	Листов
Провер.		Холкина Н. Е					4	51
Реценз.						МИ ВлГУ ПИН-121		
Н. Контр.		Холкина Н. Е.						
Утверд.		Кульков Я. Ю.						

Введение

В современном мире цифровые технологии кардинально меняют привычные модели потребления, и одним из наиболее динамично развивающихся сегментов является рынок онлайн–доставки продуктов питания. Рост популярности таких сервисов обусловлен увеличением темпа жизни, высокой занятостью населения и стремлением минимизировать временные затраты на повседневные покупки. Кроме того, пандемийные ограничения ускорили переход пользователей на цифровые каналы приобретения товаров первой необходимости, сделав онлайн–заказы не просто удобной опцией, а необходимостью.

Актуальность разработки эффективной системы онлайн–заказа и доставки продуктов питания подтверждается следующими факторами:

- ростом спроса на e–grocery. По данным исследований объем рынка онлайн продаж продуктов питания ежегодно увеличивается на 23,5%;
- изменением ожиданий потребителей. Пользователи хотят получать не только быструю доставку, но и персонализированные рекомендации, гибкие условия оплаты, а также возможность отслеживания заказа в реальном времени;
- развитием технологий – интеграция с платежными системами, использование алгоритмов машинного обучения для прогнозирования спроса и оптимизации логистики позволяют создавать более удобные и надежные сервисы;
- конкурентной средой – компании, предлагающие интуитивно понятный интерфейс, широкий ассортимент и высокий уровень сервиса, получают значительное преимущество;

Таким образом, создание удобного, функционального и безопасного сервиса онлайн–заказа продуктов питания отвечает современным рыночным трендам и потребностям пользователей, что делает данную тему крайне актуальной для исследования и разработки.

					МИВУ.09.03.04-17.000 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

1 Анализ технического задания

1.1 Описание предметной области

Проект реализует сервис по заказу и доставке продуктов питания через интернет. Пользователь выбирает продукты из каталога, формирует корзину, указывает адрес, оплачивает и подтверждает заказ. После этого заказ обрабатывается: товары резервируются, передаются в службу доставки, а пользователь получает уведомления о статусе заказа и может отслеживать доставку. По завершении клиент может оставить отзыв. Сервис также включает управление ассортиментом, логистику и персонализацию, направленные на упрощение покупок, экономию времени и повышение удобства для пользователей.

Менеджер обрабатывает заказы, управляет ассортиментом и контролирует доставку. Он просматривает и фильтрует заказы по статусу, дате или клиенту, изменяет их статусы (например, "В обработке", "Передан в доставку", "Доставлен"), вносит ручные корректировки при необходимости, связывается с клиентами в случае проблем. Менеджер также управляет каталогом продуктов: добавляет новые товары, обновляет описания, цены и остатки, скрывает отсутствующие позиции. Он координирует работу службы доставки, обрабатывает возвраты и отмены.

Курьер управляет процессом доставки: принимает назначенные заказы, обновляет их статусы ("В пути", "Доставлен"), связывается с клиентами при необходимости и обеспечивает своевременную доставку. Система позволяет курьеру видеть все активные заказы, а также архив выполненных доставок. Курьер может быть онлайн/оффлайн, что помогает администраторам эффективно распределять заказы между активными курьерами.

					МИВУ.09.03.04-17.000 ВКР	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

1.2 Особенности законодательного регулирования.

Разрабатываемая система обязана соответствовать действующему законодательству Российской Федерации, при создании учитывались такие нормативные акты, как:

1) Гражданский кодекс РФ (ст. 434, 437, 438)[1]

Что регулирует:

- заключение договоров в электронной форме (оферта);
- права и обязанности сторон при онлайн-покупках;

Требования к системе:

- публикация договора-оферты на сайте;
- чёткое описание условий оплаты, доставки, возврата;
- фиксация момента согласия с офертой (например, галочка при оформлении заказа);

2) Федеральный закон № 2300–1 "О защите прав потребителей"[2]

Что регулирует:

- права покупателей при дистанционной торговле;
- порядок возврата товаров, претензионную работу;

Требования к системе:

- предоставление полной информации о товаре (цена, характеристики, сроки доставки);

3) Федеральный закон № 152–ФЗ "О персональных данных" [3]

Что регулирует:

- сбор, хранение, обработку и защиту персональных данных (ПДн) пользователей;

- обязанности оператора по обеспечению конфиденциальности;

Требования к системе

- получение согласия пользователей на обработку данных (чекбокс при регистрации);
- шифрование и защита баз данных (SSL, хэширование паролей);

					МИВУ.09.03.04-17.000 ВКР	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

– политика конфиденциальности на сайте;

4) Условия использования API Яндекс.Карт [4]

Регулирует:

– порядок интеграции картографического сервиса в коммерческие проекты;

– ограничения на использование геоданных;

Требования к системе:

– соблюдение лицензионных ограничений при отображении карт;

– указание авторства сервиса Яндекс.Карт;

– ограничение на кэширование картографических данных;

– соблюдение технических требований к реализации API;

1.3 Обзор аналогов

Современный рынок онлайн–доставки продуктов питания представлен различными моделями работы, каждая из которых имеет свои уникальные особенности и конкурентные преимущества. Наиболее распространены две принципиально разные бизнес–стратегии: агрегаторная платформа, объединяющая существующие розничные магазины, и модель полного цикла с собственными складами и логистикой. Эти подходы предлагают потребителям разный баланс между скоростью доставки, широтой ассортимента и ценовой политикой, формируя тем самым различные ниши на рынке

1.3.1 СберМаркет

СберМаркет представляет собой классический пример агрегаторной платформы, которая объединяет множество локальных магазинов–партнеров в единую систему заказов. Основное преимущество этой модели — возможность предложить потребителям действительно широкий ассортимент товаров, так как каждый магазин–партнер привносит в систему свой

					МИВУ.09.03.04-17.000 ВКР	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

уникальный товарный портфель. Сервис интегрирован с экосистемой Сбербанка, что обеспечивает удобные финансовые сервисы и программу лояльности "Спасибо". Технологическая платформа включает мобильные приложения для пользователей и инструменты интеграции для партнеров, включая API и подключение к системам учета. Однако зависимость от работы отдельных магазинов-партнеров создает определенные сложности — в часы пик возможны задержки с обработкой заказов, а качество обслуживания может варьироваться в зависимости от конкретного магазина. Среднее время доставки составляет около 2 часов, что является приемлемым, но не рекордным показателем для рынка.

1.3.2 Яндекс.Лавка

Яндекс Лавка демонстрирует принципиально иной подход, основанный на полном контроле всей цепочки — от закупки товаров до последней мили доставки. Собственные распределенные склады и оптимизированная логистика позволяют сервису гарантировать беспрецедентно короткие сроки доставки — от 15 до 30 минут в большинстве случаев. Круглосуточный режим работы и интеграция с экосистемой Яндекса (включая подписку Плюс) создают дополнительную ценность для пользователей. Технологическая база сервиса включает современные системы прогнозирования спроса на основе искусственного интеллекта, автоматизированные складские комплексы и алгоритмы маршрутизации. Однако такая модель имеет и существенные ограничения — ассортимент товаров заметно уже, чем у агрегаторов, а необходимость содержания собственной инфраструктуры приводит к более высоким наценкам на часть товаров. География покрытия также ограничена по сравнению с агрегаторными платформами.

					МИВУ.09.03.04-17.000 ВКР	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

1.4 Анализ требований к разрабатываемой системе

Разрабатываемая система будет представлять собой комплексное решение для электронной коммерции, направленное на создание эффективной системы управления онлайн-магазином с акцентом на продажу товаров различных категорий. Система ориентирована на обеспечение удобного процесса покупок для клиентов и эффективного управления товарами для администраторов, с учетом специфики различных типов товаров и их характеристик.

Клиентский интерфейс (веб-приложение):

- персонализированные кабинеты пользователей с историей заказов и настройками профиля;
- интерактивный каталог товаров с детальным описанием характеристиками и изображением;
- система корзины покупок и оформления заказов;
- функционал отзывов и рейтингов товаров;
- система поддержки с возможностью создания запросов и обмена сообщениями;

Административная панель:

- управление товарами и категориями с гибкой системой характеристик;
- контроль заказами и пользователями;
- модерация отзывов и обращений;
- управление контентом и медиафайлами;

Мобильное приложение (интерфейс курьера):

- персонализированный профиль курьера с историей доставок и настройками;
- система управления заказами с возможностью обновления статусов;
- система уведомлений о новых заказах и изменениях статусов;

					МИВУ.09.03.04-17.000 ВКР	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

1.4.1 Функциональные требования

Требования к клиентскому интерфейсу:

- регистрация и авторизация пользователей;
- просмотр каталога товаров;
- управление корзиной покупок;
- оформление заказов с указанием адреса доставки;
- отслеживание статуса заказа;
- оставление отзывов о товарах;
- работа тех. Поддержки;
- управление личным профилем и настройками;

Требования к административной панели:

- управление пользователями и их правами;
- управление каталогом товаров и категориями;
- обработка и управление заказами;
- назначение курьеров на заказы;
- модерация отзывов и обращений;
- управление контентом и медиафайлами;
- просмотр статистики и отчетов;

Требования к мобильному приложению курьера:

- авторизация курьера;
- просмотр назначенных заказов;
- обновление статусов доставки;
- навигация к адресам доставки;
- коммуникация с клиентами;
- просмотр истории доставок;
- управление статусом доступности;

					МИВУ.09.03.04-17.000 ВКР	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

1.4.2 Нефункциональные требования

Требования к производительности:

- время отклика системы не более 4 секунд;
- поддержка одновременной работы не менее 1000 пользователей;
- обработка не менее 100 заказов в минуту;
- оптимизация работы с базой данных;

Требования к безопасности:

- защита персональных данных пользователей;
- шифрование передаваемой информации;
- JWT–авторизация (для мобильного приложения);
- защита от SQL–инъекций и XSS–атак;
- разграничение прав доступа;

Требования к мобильному приложению:

- Android нативное приложение;
- push–уведомления;

					МИВУ.09.03.04-17.000 ВКР	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

2. Проектирование и разработка системы

В данном разделе представлен комплексный подход к моделированию предметной области, проектированию базы данных проектированию архитектуры и выбору средств реализации.

Главная цель проектирования – создание гибкой и надежной архитектуры, способной поддерживать как текущие бизнес–процессы (прием и обработка заказов, управление ассортиментом, логистика), так и перспективные направления развития.

2.1 Моделирование предметной области

Моделирование предметной области представляет собой системный анализ бизнес–процессов онлайн–доставки продуктов питания и их формализацию в структуре разрабатываемой платформы. Система предназначена для комплексной автоматизации всего цикла работы сервиса – от формирования заказа клиентом до его доставки и постпродажного сопровождения. Ключевые аспекты предметной области::

Пользователи (USER): базовая роль, позволяющая совершать покупки, оставлять отзывы, добавлять товары в избранное, управлять корзиной и отправлять обращения в поддержку. Пользователи могут редактировать свой профиль, просматривать историю заказов и управлять личными настройками.

Курьеры (COURIER): верифицированные доставщики, которые помимо стандартных функций могут просматривать доступные заказы, принимать заказы на доставку

Администраторы (ADMIN): обеспечивают управление системой, обрабатывают обращения пользователей, могут управлять товарами и категориями, изменять информацию о товарах, блокировать нарушителей и удалять неприемлемый контент.

					МИВУ.09.03.04-17.000 ВКР	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

Были разработаны модели использования для каждой роли (Рисунки А.1– А.7)

2.2 Бизнес–требования

2.2.1 Цели компании/проекта

Разрабатываемая система ставит перед собой амбициозные цели по созданию эффективной и удобной платформы для онлайн–торговли, ориентированной на различные категории товаров. Система направлена на повышение вовлеченности пользователей через внедрение системы отзывов, рейтингов и персонализированных рекомендаций. Важным аспектом является создание масштабируемой системы, способной эффективно обрабатывать растущий объем заказов и пользователей.

2.2.2 Основные бизнес–функции

Система предоставит комплексный набор функций для управления онлайн–магазином. Регистрация и авторизация будут реализованы с учетом различных ролей пользователей (пользователь, администратор), что позволяет эффективно управлять профилями и историей заказов. Управление товарами включит создание и редактирование товаров с гибкой системой характеристик, категоризацию, а также контроль цен и остатков. Система управления заказами обеспечит удобное оформление заказов, отслеживание их статуса и ведение истории, сопровождаемое системой уведомлений. Административные функции предстанут удобной панелью управления, включающей модерацию товаров и отзывов, управление пользователями. Особое внимание будет уделено системе поддержки, которая позволяет создавать и обрабатывать обращения, загружать изображения и вести историю обращений.

					МИВУ.09.03.04-17.000 ВКР	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

2.2.3 Дополнительные бизнес–аспекты

В сфере правовых вопросов и безопасности веб–приложение обеспечит полное соответствие законодательству о защите персональных данных, будет проводить модерацию контента для предотвращения мошенничества и обеспечивать защиту от несанкционированного доступа. Масштабирование и развитие платформы включит поддержку растущего количества пользователей, расширение категорий товаров, интеграцию новых функций, оптимизацию производительности и расширенные возможности для администраторов.

2.3 Моделирование данных

Проведя моделирование предметной области и анализ бизнес–требований, была разработана логическая модель базы данных (Рисунок 1), отражающая все необходимые сущности и их взаимосвязи.

Модель базы данных включает следующие основные таблицы:

Пользователи и аутентификация:

- users – основная таблица пользователей;
- roles – роли пользователей (USER, ADMIN, COURIER);
- addresses – адреса пользователей;
- verification_codes – коды верификации;

Товары и категории:

- products – информация о товарах;
- categories – категории товаров;
- product_images – изображения товаров;
- product_characteristics – характеристики товаров;

Заказы и доставка:

- orders – основная информация о заказах;
- order_items – товары в заказе;
- order_statuses – статусы заказов;

					МИВУ.09.03.04-17.000 ВКР	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

- couriers – информация о курьерах;
- courier_activity – активность курьеров;

Отзывы и рейтинги:

- reviews – отзывы о товарах;
- ratings – рейтинги товаров;

Поддержка и коммуникация:

- support_tickets – обращения в поддержку;
- messages – сообщения в поддержке;
- chat_rooms – чаты между пользователями и администраторами;

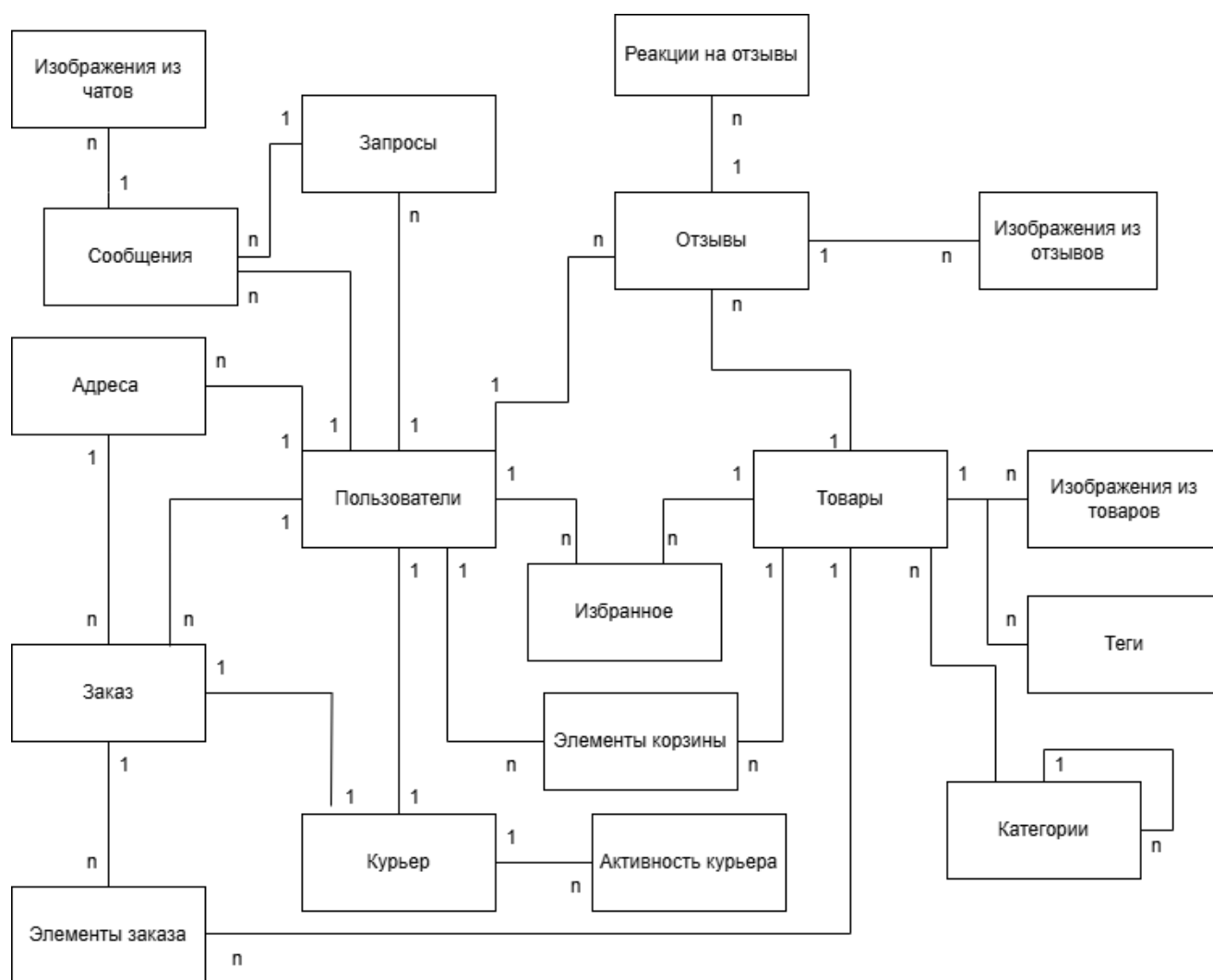


Рисунок 1 – Логическая модель базы данных

2.4 Обоснование выбора СУБД

Для разрабатываемой системы онлайн-доставки продуктов питания в качестве системы управления базами данных выбрана PostgreSQL. Данный выбор обусловлен следующими ключевыми факторами:

Надежность и стабильность.

PostgreSQL обладает доказанной надежностью и устойчивостью к сбоям, что критически важно для системы, работающей с финансовыми транзакциями и заказами пользователей. Механизм ACID-транзакций гарантирует целостность данных даже в случае системных сбоев.

Производительность.

СУБД демонстрирует высокую производительность при работе с большими объемами данных, что соответствует требованиям к системе, которая должна обрабатывать сотни заказов в час. Поддержка сложных запросов и индексов позволяет эффективно работать с каталогом товаров и историей заказов.

Масштабируемость.

PostgreSQL поддерживает вертикальное масштабирование и может быть развернута в кластерной конфигурации, что обеспечивает рост системы вместе с увеличением пользовательской базы и количества обрабатываемых заказов.

Расширяемость.

Возможность создания пользовательских функций и типов данных на различных языках программирования (включая PL/pgSQL, Python, Java) позволяет адаптировать СУБД под специфические требования проекта.

Поддержка JSON.

Встроенная работа с JSON-документами упрощает хранение и обработку переменных атрибутов товаров (характеристики, состав, пищевая ценность), а также данных пользовательских профилей.

Безопасность.

					МИВУ.09.03.04-17.000 ВКР	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

Гибкая система прав доступа и встроенные механизмы шифрования соответствуют требованиям к защите персональных данных пользователей и платежной информации.

Экономическая эффективность.

Как open-source решение, PostgreSQL не требует лицензионных отчислений, что снижает общую стоимость владения системой при сохранении профессионального уровня функциональности.

Сообщество и поддержка:

Активное сообщество разработчиков и обширная документация обеспечивают быстрый поиск решений возникающих проблем, а наличие коммерческой поддержки гарантирует стабильность работы в production-среде.

Совместимость.

Широкая поддержка различных языков программирования и фреймворков (включая Java/Spring Boot, используемые в проекте) упрощает интеграцию СУБД с остальными компонентами системы.

Выбор PostgreSQL в качестве основной СУБД для проекта обеспечивает оптимальный баланс между производительностью, надежностью, функциональностью и стоимостью владения, полностью соответствуя требованиям разрабатываемой системы онлайн-доставки продуктов питания.

2.5 Стек технологий

2.5.1 Бэкенд-технологии

В качестве основного языка программирования для разработки серверной части системы HomeShop выбран Java 21. Этот выбор обусловлен высокой производительностью благодаря JIT-компиляции и оптимизации кода, строгой типизацией, обеспечивающей надежность и предсказуемость работы приложения, богатой экосистемой библиотек и фреймворков, кроссплатформенностью благодаря JVM, поддержкой многопоточности и

					МИВУ.09.03.04-17.000 ВКР	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

асинхронного программирования, а также активным сообществом разработчиков и регулярными обновлениями. В качестве альтернатив рассматривались Python (проще в освоении, но менее производительный), Node.js (хорош для асинхронных операций, но менее надежен для сложных бизнес-процессов) и Go (высокопроизводительный, но имеет меньшую экосистему библиотек).

Spring Boot 3.4.4 выбран как основной фреймворк для разработки веб-приложения. Это решение основано на таких преимуществах как автоконфигурация, минимизирующая ручную настройку, встроенный сервер приложений, удобная система внедрения зависимостей, интеграция с различными базами данных, поддержка микросервисной архитектуры и встроенные инструменты мониторинга и безопасности. В качестве альтернатив рассматривались Django (проще в освоении, но менее гибкий), Express.js (легковесный, но требует больше ручной настройки) и Laravel (хорош для небольших проектов, но менее масштабируемый).

Для работы с базой данных используется Spring Data JPA в сочетании с Hibernate. Это решение обеспечивает упрощение доступа к данным через паттерн Repository, автоматическую генерацию SQL-запросов, поддержку транзакций и кэширования, маппинг объектов на таблицы базы данных и оптимизацию производительности. Альтернативными решениями могли быть MyBatis (более гибкий, но требует больше ручной работы), JOOQ (типобезопасный, но менее абстрактный) и JDBC (прямой доступ к базе данных, но требует больше кода).

Безопасность системы обеспечивается с помощью Spring Security, который предоставляет JWT-аутентификацию, ролевую модель доступа, шифрование паролей и интеграцию с OAuth2. В качестве альтернатив рассматривались Apache Shiro (проще в настройке, но менее функциональный) и Custom Security (полный контроль, но требует больше разработки).

					МИВУ.09.03.04-17.000 ВКР	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

Для валидации данных используется Jakarta Validation API, который обеспечивает декларативную валидацию через аннотации, поддержку кастомных валидаторов, интеграцию с Spring и проверку на стороне сервера. Альтернативными решениями могли быть Hibernate Validator (более функциональный, но специфичный), Custom Validation (полный контроль, но требует больше кода) и Bean Validation (стандартный подход, но менее гибкий).

Обработка JSON-данных осуществляется с помощью Jackson, который обеспечивает высокую производительность, поддержку различных форматов данных, гибкую настройку сериализации/десериализации и интеграцию с Spring. В качестве альтернатив рассматривались Gson (проще в использовании, но менее производительный), JSON-B (стандартный подход, но менее гибкий) и Custom JSON Parser (полный контроль, но требует больше разработки). Выбор именно этого стека технологий для бэкенда обусловлен их надежностью, производительностью и широкой поддержкой сообществом разработчиков. Каждая технология была выбрана с учетом специфических требований проекта и необходимости обеспечения масштабируемости системы.

2.5.2 Фронтенд

Для разработки пользовательского интерфейса системы выбран комплексный подход с использованием Thymeleaf, Tailwind CSS, Alpine.js и JavaScript. Этот выбор обусловлен необходимостью создания современного, отзывчивого и удобного интерфейса для пользователей.

Thymeleaf выбран в качестве основного шаблонизатора на стороне сервера. Это решение основано на таких преимуществах как естественная интеграция с Spring Boot, возможность использования HTML-шаблонов без необходимости компиляции, поддержка фрагментов для переиспользования кода и удобная работа с данными на стороне сервера. В качестве альтернатив

					МИВУ.09.03.04-17.000 ВКР	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

рассматривались JSP (устаревшая технология), FreeMarker (менее гибкий) и Mustache (ограниченная функциональность).

Для стилизации интерфейса используется Tailwind CSS– утилитарный CSS–фреймворк, который позволяет создавать современные и отзывчивые интерфейсы. Выбор Tailwind CSS обусловлен его гибкостью, отсутствием необходимости писать собственные CSS–классы, поддержкой адаптивного дизайна и высокой производительностью. Альтернативными решениями могли быть Bootstrap (более тяжелый, менее гибкий), Material–UI (привязан к Material Design) и собственные CSS–стили (требуют больше времени на разработку).

Alpine.js используется для добавления интерактивности в HTML–шаблоны. Этот легковесный JavaScript–фреймворк (всего 7.1kB) позволяет создавать динамические компоненты прямо в HTML, что значительно упрощает разработку и делает код более понятным. В системе Alpine.js применяется для:

- управления состоянием компонентов;
- обработки событий;
- реализации выпадающих меню;
- управления модальными окнами;
- валидации форм;
- анимации и переходов;

В качестве альтернатив Alpine.js рассматривались Vue.js (более тяжелый, требует отдельной сборки) и jQuery (устаревший подход), но Alpine.js был выбран за его простоту интеграции с Thymeleaf и минимальные накладные расходы.

JavaScript используется для реализации более сложной интерактивности и асинхронных запросов к серверу. Это позволяет создавать динамический пользовательский интерфейс без перезагрузки страницы, реализовывать валидацию форм на стороне клиента и обеспечивать плавные анимации. В

					МИВУ.09.03.04-17.000 ВКР	Лист
						21
Изм.	Лист	№ докум.	Подпись	Дата		

качестве альтернатив рассматривались TypeScript (добавляет типизацию, но усложняет разработку) и CoffeeScript (упрощает синтаксис, но менее распространен).

Особое внимание уделено адаптивному дизайну, который обеспечивает корректное отображение интерфейса на различных устройствах. Это достигается с помощью комбинации Tailwind CSS и медиа-запросов, что позволяет создавать интерфейс, который одинаково хорошо работает как на десктопных, так и на мобильных устройствах.

Для улучшения пользовательского опыта реализованы:

- асинхронная загрузка данных без перезагрузки страницы;
- валидация форм на стороне клиента;
- динамическое обновление корзины и списка избранного;
- плавные анимации и переходы;
- адаптивные изображения и медиа-контент;

Выбор именно этого стека технологий для фронтенда обусловлен их надежностью, производительностью и удобством разработки. Каждая технология была выбрана с учетом специфических требований проекта и необходимости обеспечения качественного пользовательского опыта.

2.5.3 Мобильное приложение

Для разработки мобильного приложения курьера выбран Kotlin 2.0.21 в качестве основного языка программирования. Этот выбор обусловлен современным синтаксисом, null-safety системой, поддержкой корутин для асинхронного программирования, полной совместимостью с Java и активной поддержкой от Google. В качестве альтернатив рассматривались Java (более verbose, менее современный), Flutter (кроссплатформенность, но менее нативная производительность) и React Native (веб-технологии, но менее нативная интеграция).

					МИВУ.09.03.04-17.000 ВКР	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

Архитектура приложения построена на основе MVVM (Model–View–ViewModel) с использованием компонентов Android Architecture Components. Это решение обеспечивает разделение ответственности между слоями, упрощает тестирование, обеспечивает сохранение состояния при изменениях конфигурации и следует принципам чистой архитектуры. В качестве альтернатив рассматривались MVC (менее гибкий), MVP (более verbose) и MVI (сложнее в реализации).

Для внедрения зависимостей используется Hilt 2.48, который предоставляет удобный способ управления зависимостями, упрощает тестирование и обеспечивает единую точку конфигурации. Альтернативными решениями могли быть Koin (легковесный, но менее типобезопасный), Dagger (более verbose) и Kodein (менее распространенный).

Сетевой слой реализован с использованием Retrofit 2.9.0 в сочетании с OkHttp 4.12.0. Это решение обеспечивает типобезопасные HTTP–запросы, автоматическую сериализацию/десериализацию JSON, поддержку корутин, удобное логирование запросов и обработку ошибок. В качестве альтернатив рассматривались Ktor (менее распространенный), Volley (устаревший) и собственные решения (требуют больше разработки).

Для работы с данными используется Gson 2.10.1, который обеспечивает эффективную сериализацию/десериализацию JSON, поддержку кастомных типов и удобную работу с null–значениями. Альтернативными решениями могли быть Moshi (более строгая типизация, но менее гибкий), Jackson (более тяжелый) и Kotlinx Serialization (менее зрелый).

UI компоненты реализованы с использованием Material Design 3 (версия 1.11.0), что обеспечивает современный внешний вид, соответствие гайдлайнам Material Design и удобные готовые компоненты. Для навигации между экранами используется Navigation Component 2.7.7, который предоставляет типобезопасную навигацию, поддержку глубоких ссылок и анимации переходов.

					МИВУ.09.03.04-17.000 ВКР	Лист
						23
Изм.	Лист	№ докум.	Подпись	Дата		

Для работы с датами используется ThreeTenABP 1.4.6, который обеспечивает удобную работу с датами и временем, поддержку временных зон и форматирование. В качестве альтернатив рассматривались Joda-Time (устаревший) и собственные решения (требуют больше разработки).

Особое внимание уделено безопасности приложения:

- безопасное хранение JWT токена;
- SSL/TLS для шифрования трафика;
- защита от перехвата данных;
- проверка подписи приложения;
- безопасное хранение чувствительных данных;

Приложение поддерживает Android версии от 8.0 (API 26) до Android 14 (API 34), что обеспечивает охват более 95% активных Android-устройств. Для обеспечения производительности и отзывчивости интерфейса используются:

- ViewBinding для типобезопасного доступа к view;
- Lifecycle компоненты для управления жизненным циклом;
- LiveData для реактивного обновления UI;
- корутины для асинхронных операций;
- оптимизация памяти и батареи;

Выбор именно этого стека технологий для мобильного приложения обусловлен их надежностью, производительностью и удобством разработки. Каждая технология была выбрана с учетом специфических требований проекта и необходимости обеспечения качественного пользовательского опыта для курьеров.

2.5.4 Конфигурация безопасности

Система будет построена на базе Spring Security с двумя основными конфигурациями:

Основная веб-конфигурация (@Order(2)):

- защита всех веб-эндпоинтов;

					МИВУ.09.03.04-17.000 ВКР	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

- форма–аутентификация;
- управление сессиями;
- ролевая модель доступа;

Мобильная конфигурация (@Order(1)):

- отдельная конфигурация для мобильного API;
- JWT–аутентификация;
- статус–независимая сессия;
- аутентификация;

Веб–интерфейс использует классическую форму аутентификации с управлением сессиями. Пользователи проходят аутентификацию через форму входа на странице /auth/login, после успешного входа происходит перенаправление на главную страницу. Система обеспечивает защиту от брутфорс–атак путем ограничения количества активных сессий до одной на пользователя, что предотвращает параллельные попытки входа. При выходе из системы происходит полное удаление сессии и cookies. Для мобильного приложения реализована отдельная система безопасности на основе JWT (JSON Web Tokens). Это позволяет обеспечить статус–независимую аутентификацию, что особенно важно для мобильных приложений. Курьеры имеют отдельный эндпоинт для входа и специальную роль COURIER, что обеспечивает изолированный доступ к API курьера.

Система авторизации построена на основе ролевой модели, где определены три основные роли: ADMIN для администраторов, USER для обычных пользователей и COURIER для курьеров. Каждая роль имеет четко определенный набор доступных ресурсов. Публичные ресурсы, такие как страницы аутентификации, регистрации и статические файлы, доступны всем пользователям. Административные ресурсы защищены ролью ADMIN, пользовательские функции доступны только авторизованным пользователям с

					МИВУ.09.03.04-17.000 ВКР	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

ролью USER, а API курьера доступно только для пользователей с ролью COURIER.

Безопасность соединения обеспечивается принудительным использованием HTTPS для всех запросов. CSRF-защита отключена для API-эндпоинтов, но активна для веб-интерфейса. Статические ресурсы обрабатываются через специальный ResourceHandler, что обеспечивает безопасный доступ к загруженным файлам.

Система включает комплексную обработку ошибок с кастомными страницами для веб-интерфейса и JSON-ответами для API. Все ошибки аутентификации логируются для последующего анализа. Пароли пользователей хранятся в зашифрованном виде с использованием BCrypt, что обеспечивает надежную защиту учетных данных.

Такая архитектура безопасности обеспечивает надежную защиту всех компонентов системы, при этом сохраняя гибкость и масштабируемость. Разделение конфигураций для веб- и мобильного интерфейсов позволяет независимо развивать и настраивать безопасность для каждого типа клиентов, что особенно важно в современных условиях, когда система должна обслуживать различные типы устройств и приложений.

2.6 Физическая модель базы данных

Физическая модель базы данных (Рисунки А.8–А.9) разработана с учетом требований к производительности и надежности системы. В модели реализованы:

- конкретные типы данных для всех полей таблиц;
- ограничения целостности (первичные и внешние ключи);
- транзакционные механизмы обработки данных;
- оптимальная структура таблиц в третьей нормальной форме;

Модель обеспечивает:

- минимальную избыточность хранимых данных;

					МИВУ.09.03.04-17.000 ВКР	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

- поддержку целостности при параллельном доступе;
- эффективное выполнение запросов;
- возможность масштабирования при росте нагрузки;

					МИВУ.09.03.04-17.000 ВКР	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

3 Описание системы

Разработанная система для управления доставкой включает веб-приложение для администрирования и мобильное приложение для курьеров. Веб-часть позволяет управлять заказами, контролировать выполнение и анализировать работу. Мобильное приложение дает курьерам доступ к статусам заказов и инструментам подтверждения доставки. Данные между платформами синхронизируются в реальном времени, обеспечивая актуальность информации.

3.1 Руководство программиста

Этот раздел содержит подробное описание всех пакетов и классов проекта, включая информацию об их предназначении, основных методах и свойствах, а также взаимосвязях с другими компонентами системы.

Пакет "Utils" включает классы, отвечающие за конфигурацию приложения, так же содержит вспомогательные классы.

1) Класс SecurityConfig (Конфигурация безопасности).

Основные методы:

- addResourceHandlers(ResourceHandlerRegistry registry): void – настраивает обработку статических ресурсов;
- authenticationManager(AuthenticationManagerBuilder auth): void – создает менеджер аутентификации;
- webSecurityFilterChain(HttpSecurity http): SecurityFilterChain – настраивает цепочку фильтров безопасности для веб-приложения;
- apiFilterChain(HttpSecurity http): SecurityFilterChain – настраивает цепочку фильтров безопасности для API-запросов;

2) Класс JwtUtil (Утилита для работы с JWT токенами).

Основные методы:

- generateToken(String email, String role): String – генерирует JWT токен;

					МИВУ.09.03.04-17.000 ВКР	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

- createToken(Map<String, Object> claims, String subject): String – создает токен с claims;
- getClaimsFromToken(String token): Claims – получает claims из токена;
- getEmailFromToken(String token): String – извлекает email из токена;
- validateToken(String token, String email): Boolean – проверяет валидность токена;

- isTokenExpired(String token): boolean – проверяет истек ли токен;

3) Класс ProductImageComparator (Компаратор для сортировки изображений продуктов).

Основные методы:

- compare(ProductImage a, ProductImage b): int – сравнивает изображения продуктов. Сортирует сначала по primary (true идет первым), затем по id;

4) Класс CustomAuthenticationProvider (Провайдер аутентификации).

Основные методы:

- authenticate(Authentication authentication): Authentication – выполняет аутентификацию;
- supports(Class<?> authentication): boolean – проверяет поддержку типа аутентификации.;

5) Класс JwtAuthenticationFilter (Фильтр для JWT аутентификации).

Основные методы:

- doFilterInternal(HttpServletRequest request, HttpServletResponse response, FilterChain chain): void;

Основные особенности пакета Utils:

- содержит утилиты для работы с JWT токенами;
- реализует механизмы безопасности и аутентификации;
- предоставляет инструменты для работы с изображениями продуктов;
- настраивает SSL и HTTPS;
- обеспечивает безопасную обработку запросов;
- реализует кастомную аутентификацию;

– управляет доступом к ресурсам;

Пакет "Controllers" включает классы, отвечающие http-запросы.

1) Класс MainController (Отвечает за главную страницу)

Основные методы:

– mainPage() – отображает главную страницу приложения, определяет роль пользователя и показывает соответствующий интерфейс (обычный пользователь или администратор);

– addToCart() – добавляет выбранный товар в корзину пользователя, проверяет наличие товара и обновляет количество;

– removeFromCart() – удаляет товар из корзины пользователя;

– updateCartItem() – обновляет количество товара в корзине, проверяет доступное количество;

2) Класс AuthController (Контроллер управления аутентификацией и регистрацией пользователей. Обеспечивает безопасный вход в систему, регистрацию новых пользователей и управление их учетными данными. Включает механизмы валидации и проверки безопасности).

Основные методы:

– showLoginPage() – отображает страницу входа в систему, обрабатывает параметры ошибок и сообщений;

– showRegistrationForm() – показывает форму регистрации нового пользователя;

– registerUser() – обрабатывает регистрацию нового пользователя, проверяет данные и создает аккаунт;

– validatePassword() – проверяет пароль на соответствие требованиям безопасности;

– isCommonPassword() – проверяет, не является ли пароль слишком распространенным;

– isPasswordSimilarToPersonalData() – проверяет, не содержит ли пароль личные данные пользователя;

					МИВУ.09.03.04-17.000 ВКР	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

3) Класс ProductController. Контроллер управления товарами в системе. Обеспечивает отображение каталога товаров, поиск, фильтрацию и просмотр детальной информации о продуктах. Включает функционал пагинации и сортировки.

Основные методы:

- showProductCatalog() – отображает каталог товаров с возможностью пагинации и фильтрации;
- getProductDetails() – показывает детальную информацию о конкретном товаре;
- searchProducts() – выполняет поиск товаров по названию или описанию;
- filterProducts() – фильтрует товары по различным параметрам (категория, цена, наличие) ;

4) Класс CartController (Контроллер управления корзиной покупок. Отвечает за все операции с корзиной: добавление товаров, изменение количества, удаление и расчет общей стоимости. Включает проверки доступности товаров и лимитов).

Основные методы:

- viewCart() – отображает содержимое корзины пользователя;
- addToCart() – добавляет товар в корзину с указанным количеством;
- updateQuantity() – изменяет количество товара в корзине;
- removeItem() – удаляет товар из корзины;
- calculateTotal() – рассчитывает общую стоимость товаров в корзине;

5) Класс OrderController (Контроллер управления заказами пользователей. Обеспечивает создание, просмотр и управление заказами. Включает функционал отслеживания статуса заказа и интеграцию с курьерской службой).

Основные методы:

- createOrder() – создает новый заказ на основе товаров в корзине;

					МИВУ.09.03.04-17.000 ВКР	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		

- viewOrders() – показывает список заказов пользователя;
- cancelOrder() – отменяет существующий заказ;
- trackOrder() – отображает текущий статус заказа;
- updateOrderStatus() – обновляет статус заказа;

6) Класс UserController (Контроллер управления профилем пользователя. Отвечает за отображение и редактирование личной информации пользователя, управление настройками аккаунта и просмотр истории заказов).

Основные методы:

- showProfile() – отображает профиль пользователя;
- updateProfile() – обновляет информацию в профиле;
- changePassword() – меняет пароль пользователя;
- viewOrderHistory() – показывает историю заказов;

Пакет "Admin" внутри пакета контроллеров реализует запросы от администраторов

7) Класс UserAdminController (Контроллер административного управления пользователями. Обеспечивает полный контроль над пользователями системы: просмотр, редактирование, блокировку и удаление аккаунтов).

Основные методы:

- showUsersPage() – отображает страницу управления пользователями
- updateUserRole() – изменяет роль пользователя в системе
- toggleUserEnabled() – включает/отключает аккаунт пользователя
- deleteUser() – удаляет пользователя из системы

8) Класс ProductCRUDController (Контроллер административного управления товарами. Обеспечивает полный цикл управления товарами: создание, чтение, обновление и удаление (CRUD операции)).

Основные методы:

- showProductList() – показывает список всех товаров;
- showAddProductForm() – отображает форму добавления нового товара;

- addProduct() – добавляет новый товар в систему;
- editProduct() – редактирует существующий товар;
- deleteProduct() – удаляет товар из системы;

9) Класс CategoryController (Контроллер управления категориями товаров. Отвечает за создание и управление структурой категорий, включая подкатегории и их взаимосвязи).

Основные методы:

- listCategories() – отображает список категорий товаров;
- showAddCategoryForm() – показывает форму добавления категории;
- addCategory() – создает новую категорию;
- editCategory() – редактирует существующую категорию;
- deleteCategory() – удаляет категорию;

10) Класс OrderAdminController (Контроллер административного управления заказами. Обеспечивает полный контроль над заказами в системе, включая назначение курьеров и управление статусами).

Основные методы:

- showOrders() – отображает список всех заказов;
- updateOrderStatus() – изменяет статус заказа;
- assignCourier() – назначает курьера на заказ;
- viewOrderDetails() – показывает детальную информацию о заказе;

Пакет "Courier" внутри пакета контроллеров реализует запросы от курьеров.

1) Класс CourierAuthController (Контроллер аутентификации и управления заказами для курьеров. Обеспечивает безопасный вход в систему для курьеров и управление назначенными заказами).

Основные методы:

- login() – аутентификация курьера в системе;
- getAssignedOrders() – получение списка назначенных заказов;
- updateOrderStatus() – обновление статуса доставки заказа;

– completeOrder() – завершение доставки заказа;

Пакет "Services" включает классы, отвечающие за бизнес-логику.

1) Класс UserService (Сервис управления пользователями. Отвечает за все операции с пользователями, включая аутентификацию, регистрацию и управление профилями).

Основные методы:

– getCurrentUser() – получает текущего аутентифицированного пользователя;

– registerUser() – регистрирует нового пользователя;

– findByEmail() – находит пользователя по email;

– updateUser() – обновляет информацию о пользователе;

– changePassword() – меняет пароль пользователя;

– findUsersByRole() – находит пользователей по роли;

– toggleUserEnabled() – включает/отключает аккаунт пользователя;

– deleteUser() – удаляет пользователя;

Класс ProductService (Сервис управления товарами. Обеспечивает все операции с товарами, включая создание, обновление, поиск и фильтрацию).

Основные методы:

– findAll() – получает все товары;

– findById() – находит товар по ID;

– saveProduct() – сохраняет новый товар;

– updateProduct() – обновляет существующий товар;

– deleteProduct() – удаляет товар;

– searchProducts() – ищет товары по параметрам;

– getProductsByCategory() – получает товары по категории;

– getRandomProducts() – получает случайные товары;

– updateStock() – обновляет количество товара;

2) Класс CartService (Сервис управления корзиной покупок. Обработывает все операции с корзиной и товарами в ней).

					МИВУ.09.03.04-17.000 ВКР	Лист
						34
Изм.	Лист	№ докум.	Подпись	Дата		

Методы:

- getCartItems() – получает товары в корзине;
- addToCart() – добавляет товар в корзину;
- updateQuantity() – обновляет количество товара;
- removeFromCart() – удаляет товар из корзины;
- clearCart() – очищает корзину;
- calculateTotal() – рассчитывает общую стоимость;
- getCartItemsCount() – получает количество товаров;
- convertToKg() – конвертирует вес в килограммы;
- isWeightLimitExceeded() – проверяет превышение лимита веса;

3) Класс OrderService (Сервис управления заказами. Обеспечивает создание и управление заказами, их статусами и доставкой).

Основные методы:

- createOrder() – создает новый заказ;
- findOrderById() – находит заказ по ID;
- getUserOrders() – получает заказы пользователя;
- updateOrderStatus() – обновляет статус заказа;
- cancelOrder() – отменяет заказ;
- assignCourier() – назначает курьера;
- getOrdersByStatus() – получает заказы по статусу;
- calculateOrderTotal() – рассчитывает стоимость заказа;

4) Класс CategoryService (Сервис управления категориями товаров. Обеспечивает работу с категориями и их иерархией).

Основные методы:

- getAllCategories() – получает все категории;
- findById() – находит категорию по ID;
- saveCategory() – сохраняет новую категорию;
- updateCategory() – обновляет категорию;
- deleteCategory() – удаляет категорию;

					МИВУ.09.03.04-17.000 ВКР	Лист
						35
Изм.	Лист	№ докум.	Подпись	Дата		

- getRootCategories() – получает корневые категории;
- getSubcategories() – получает подкатегории;
- getCategoryTree() – получает дерево категорий;

5) Класс ReviewService (Сервис управления отзывами о товарах.

Обработывает создание и управление отзывами).

Основные методы:

- createReview() – создает новый отзыв;
- findByProductId() – находит отзывы по товару;
- updateReview() – обновляет отзыв;
- deleteReview() – удаляет отзыв;
- getAverageRating() – получает средний рейтинг;
- validateReview() – проверяет отзыв;

6) Класс CourierService (Сервис управления курьерами и их заказами.

Обеспечивает работу с курьерами и их доставками).

Основные методы:

- createCourier() – создает нового курьера;
- findCourierById() – находит курьера по ID;
- updateCourier() – обновляет информацию о курьере;
- getAssignedOrders() – получает назначенные заказы;
- completeOrder() – завершает доставку заказа;

7) Класс ChatService (Сервис управления чатами и сообщениями.

Обеспечивает коммуникацию между пользователями и поддержкой).

Основные методы:

- createTicket() – создает новый тикет;
- sendMessage() – отправляет сообщение;
- getMessages() – получает сообщения;
- updateTicketStatus() – обновляет статус тикета;
- getOpenTickets() – получает открытые тикеты;
- closeTicket() – закрывает тикет;

					МИВУ.09.03.04-17.000 ВКР	Лист
						36
Изм.	Лист	№ докум.	Подпись	Дата		

8) Класс EmailService (Сервис отправки электронных писем. Обеспечивает коммуникацию с пользователями через email).

Основные методы:

- sendConfirmationEmail() – отправляет письмо подтверждения;
- sendPasswordResetEmail() – отправляет письмо сброса пароля;
- sendOrderConfirmation() – отправляет подтверждение заказа;
- sendOrderStatusUpdate() – отправляет обновление статуса заказа;

9) Класс FileStorageService (Сервис управления файлами. Обеспечивает загрузку и хранение файлов).

Основные методы:

- store() – сохраняет файл;
- load() – загружает файл;
- delete() – удаляет файл;
- getFileUrl() – получает URL файла ;

Пакет "POJO" включает классы сущностей, которые представляют основные бизнес-объекты приложения.

Пакет "DTO" включает классы для передачи данных между различными слоями приложения.

Пакет "Repository" включает интерфейсы и классы для взаимодействия с базой данных и выполнения CRUD операций.

3.2 Руководство пользователя

3.2.1 Регистрация и авторизация

При посещении портала пользователю доступна страница входа в систему и основная страница на которой можно посмотреть некоторые товары и перейти на страницу авторизации. На странице для авторизации в центре экрана расположена форма авторизации с полями для ввода данных. Верхнее поле предназначено для ввода email, ниже находится поле для ввода пароля (Рисунок 1). Под полями ввода расположена кнопка "Войти", нажатие на

					МИВУ.09.03.04-17.000 ВКР	Лист
						37
Изм.	Лист	№ докум.	Подпись	Дата		

которую отправляет введенные данные на проверку. Если авторизация прошла успешно, пользователь будет автоматически перенаправлен на главную страницу каталога товаров. В случае ошибки (неправильный email или пароль) над формой появится текстовое сообщение с предупреждением (Рисунок Б,1).

Вход в аккаунт

Email адрес
onze1337@mail.ru

Пароль
.....

Войти

Ещё нет аккаунта? [Зарегистрируйтесь](#)

Рисунок 2 – Форма авторизации

3.2.2 Главная страница

После успешного входа в систему пользователю становится доступна страница актуальных товаров (Рисунок 4). В верхней части страницы расположено навигационное меню, содержащее следующие разделы:

- Товары;
- Избранное;
- Заказы;
- Профиль;
- Корзина;

Основное пространство страницы занимает сетка товарных карточек. Каждая карточка содержит:

- Изображение товара;
- Название продукции;
- Вес;
- Оценку;
- Цену;
- Кнопки взаимодействия;

Шестым блоком является переход на страницу с полным каталогом и продвинутым поиском

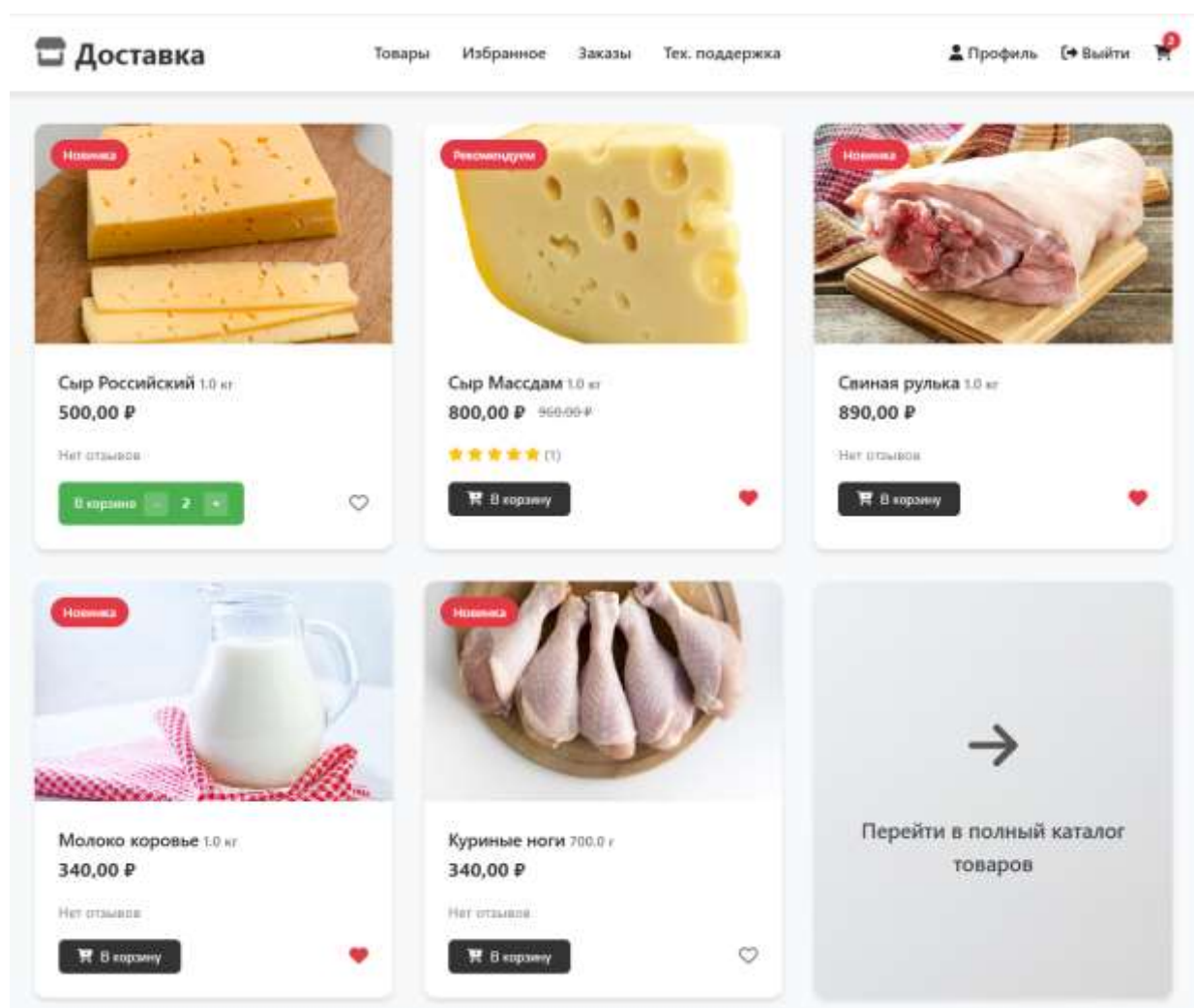


Рисунок 3 – Основная страница

3.2.3 Корзина

Перейти на страницу корзины можно через иконку корзины, расположенную в правом верхнем углу навигационного меню.

Здесь отображается общее количество товаров, общая стоимость заказа и общий вес всех товаров (Рисунок 5). Эта информация автоматически обновляется при любых изменениях в корзине.

Основное пространство страницы занимает список товаров, где каждый товар представлен в виде отдельной карточки. Карточка товара содержит фотографию товара, его название, основные характеристики (вес, объем, количество в упаковке), цену за единицу и общую стоимость позиции с учетом количества. Под информацией о товаре расположены элементы управления количеством: кнопка "—" для уменьшения количества, поле с текущим количеством и кнопка "+" для увеличения. При нажатии на кнопки "+" и "—" количество товара изменяется на единицу, а общая стоимость позиции и итоговая сумма заказа пересчитываются автоматически.

Справа от элементов управления количеством находится кнопка удаления товара из корзины, при нажатии на которую товар немедленно удаляется из списка. При удалении товара система автоматически пересчитывает общую стоимость заказа и общий вес.

Под списком товаров расположен блок с итоговой информацией о заказе, где отображается:

- Общее количество товаров;
- Общая стоимость заказа;
- Общий вес всех товаров;

В нижней части страницы находится кнопка "Оформить заказ", при нажатии на которую пользователь переходит к процессу оформления заказа. Если корзина пуста, вместо списка товаров отображается сообщение "Ваша корзина пуста" с кнопкой перехода в каталог товаров.

					МИВУ.09.03.04-17.000 ВКР	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

При наведении курсора на название товара оно становится активным, и при клике происходит переход на страницу с детальной информацией о товаре, где можно просмотреть полное описание, все характеристики, отзывы других покупателей и дополнительные фотографии.

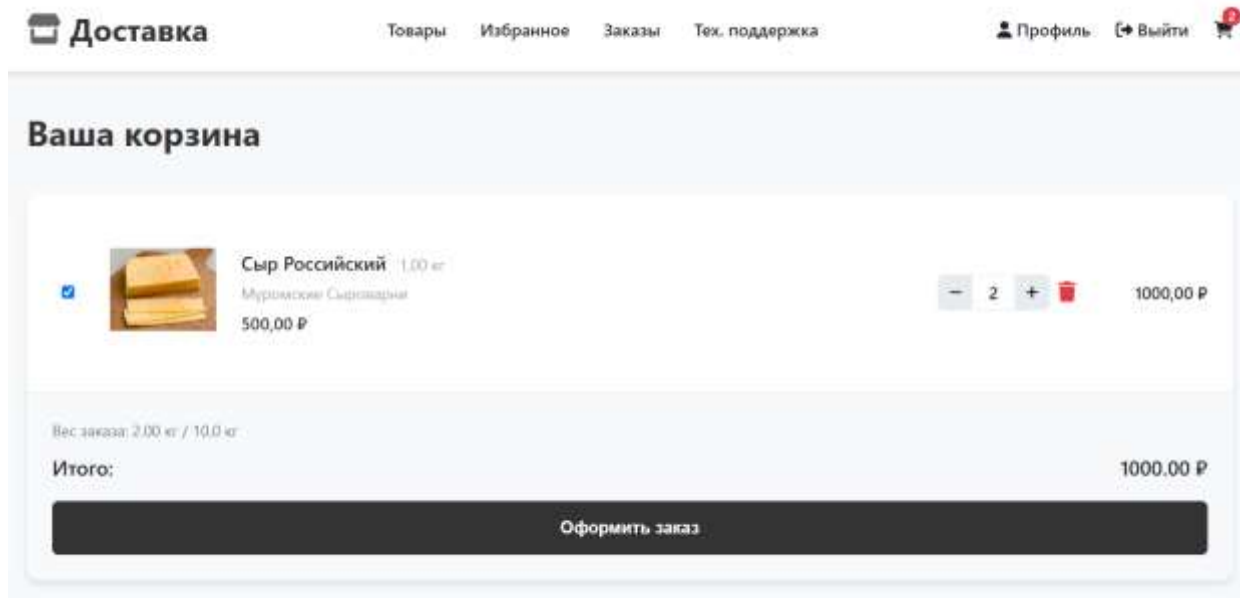


Рисунок 4 – Корзина

После нажатия кнопки "Оформить заказ" в корзине пользователь переходит на страницу оформления заказа (Рисунок 6). Страница разделена на несколько логических блоков, каждый из которых содержит важную информацию и элементы управления.

В первом блоке "Информация о доставке" пользователь может выбрать адрес доставки из списка ранее сохраненных адресов или добавить новый адрес. При выборе адреса система автоматически определяет доступные даты и время доставки.

В нижней части страницы расположены две кнопки:

- "Подтвердить заказ"— при нажатии система обрабатывает заказ и переводит пользователя на страницу подтверждения;
- "Вернуться в корзину"— позволяет вернуться к редактированию состава заказа;

После нажатия кнопки "Подтвердить заказ" система проверяет все введенные данные, резервирует товары на складе и создает заказ.

					МИВУ.09.03.04-17.000 ВКР	Лист
						41
Изм.	Лист	№ докум.	Подпись	Дата		

Пользователь получает уведомление об успешном оформлении заказа с его номером, который можно использовать для отслеживания статуса доставки.

Доставка

ТоварыИзбранноеЗаказыТех. поддержка

ПрофильВыйти

Оформление заказа

Ваша последняя адрес:

Орловская улица, 23

Адрес доставки в Муроме:

Владимирская область, Муром, Орловская улица, 1

Адрес или объект

Найти

Пробег

Слон

Как добраться

Доехать на такси

Создать свою карту

50 м

© Яндекс УСЛОВИЯ ИСПОЛЬЗОВАНИЯ

Способ оплаты:

Наличными

Подтвердить заказ

Вернуться в корзину

Рисунок 5 – Оформление заказа

3.2.5 Заказы

Раздел "Заказы" доступен через навигационное меню в верхней части страницы. При переходе в этот раздел пользователь видит полный список всех своих заказов, отсортированных по дате создания (от новых к старым). В первом блоке отображаются активные заказы, во-втором завершённые (Рисунок Б, 2).

Каждый заказ в списке представлен в виде карточки, содержащей основную информацию:

- Номер заказа (уникальный идентификатор);
- Дата и время создания заказа;
- Текущий статус заказа (выделен соответствующим цветом для удобства восприятия);
- Общая стоимость заказа;
- Количество товаров в заказе;

При наведении курсора на карточку заказа она подсвечивается, а при клике происходит переход на страницу с детальной информацией о заказе. На странице деталей заказа отображается полная информация:

– В блоке "Информация о заказе":

- Номер заказа;
- Дата и время создания;
- Текущий статус;
- Способ оплаты;
- Адрес доставки;

В блоке "Состав заказа":

- Полный список товаров с фотографиями;
- Количество каждого товара;
- Цена за единицу;
- Общая стоимость каждой позиции;

3.2.6 Профиль пользователя

Раздел "Профиль" доступен через иконку пользователя в верхнем навигационном меню. При переходе в профиль пользователь видит страницу, разделенную на несколько основных блоков (Рисунок Б, 3). В блоке "Личная информация" отображаются и могут быть отредактированы: имя пользователя, email, телефон, дата рождения, пол. Все поля имеют валидацию, и система проверяет корректность введенных данных.

					МИВУ.09.03.04-17.000 ВКР	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

В блоке "История заказов" отображается список всех заказов пользователя с возможностью быстрого перехода к деталям любого заказа. Заказы сгруппированы по месяцам для удобства навигации.

Ниже отображаются несколько продуктов из избранных

3.2.7 Чат с поддержкой

Раздел "Поддержка" доступен через навигационное меню. При переходе в этот раздел пользователь видит список своих обращений и интерфейс чата. В списке отображаются все обращения с их статусом (Открыто/В обработке/Закрыто) и датой создания. При выборе обращения открывается чат с историей переписки, где сообщения пользователя и службы поддержки визуально различаются. Пользователь может создавать новые обращения, отправлять сообщения и прикреплять до 2 изображений (jpg, jpeg, png) размером до 5 МБ каждый.

3.2.8 Отзывы и рейтинги

На странице товара в разделе "Отзывы" пользователи могут оставлять свои оценки и комментарии (Рисунок 3). Форма отзыва содержит: поле для текста отзыва, оценку по 5-балльной шкале (звезды), возможность прикрепления фотографий товара. Перед публикацией отзыва система проверяет, был ли товар приобретен пользователем

Все отзывы отображаются в порядке рейтинга с указанием: даты публикации, имени пользователя, оценки, текста отзыва и фотографий. Под каждым отзывом есть кнопки "Полезно" и "Не полезно", позволяющие оценить полезность отзыва. Система подсчитывает количество положительных и отрицательных оценок и отображает их рядом с отзывом.

					МИВУ.09.03.04-17.000 ВКР	Лист
						44
Изм.	Лист	№ докум.	Подпись	Дата		

Отзывы

Средняя оценка: 5,0 (1 отзыв)

Саша

14.06.2025 05:15



Великолепно!



+ 1 -

Оставить отзыв



Пожалуйста, поставьте оценку:

Напишите ваш отзыв...

☐ Оставить отзыв анонимно

Фотографии (необязательно, до 5 штук):

Выбрать файлы

Файл не выбран

Поставьте оценку

Рисунок 6 – Раздел с отзывами

3.2.9 Избранное

Раздел "Избранное" доступен через навигационное меню. В этом разделе отображаются все товары, которые пользователь добавил в избранное. Каждая карточка товара содержит: фотографию, название, текущую цену, кнопку быстрого добавления в корзину.

Товары в избранном можно сортировать. При наведении на товар появляется кнопка удаления из избранного.

					МИВУ.09.03.04-17.000 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		45

3.3 Руководство администратора

3.2.2 Панель управления администратора

После успешного входа администратор получает доступ к панели управления (Рисунок Б.4), которая содержит следующие основные разделы:

- Товары – управление каталогом товаров;
- Категории – управление категориями товаров;
- Пользователи – управление пользователями системы;
- Поддержка – управление обращениями пользователей;
- Заказы – управление заказами;
- Курьеры – управление курьерами;

Каждый раздел представлен в виде карточки с соответствующей иконкой и названием. При наведении на карточку она подсвечивается, а при клике происходит переход в соответствующий раздел управления.

3.2.3 Управление товарами

Раздел "Товары"(Рисунок 4) предоставляет администратору полный контроль над каталогом товаров. Здесь можно просматривать полный список всех товаров с детальной информацией о каждом: название, категория, цена, наличие, статус и другие параметры. Для добавления нового товара администратор использует специальную форму, которая включает в себя все необходимые поля: основную информацию (название, цена, вес), выбор категории, загрузку изображений, характеристики товара и подробное описание.

При редактировании существующего товара администратор может изменить любые параметры, включая цену, наличие, описание и

					МИВУ.09.03.04-17.000 ВКР	Лист
						46
Изм.	Лист	№ докум.	Подпись	Дата		

характеристики. Система автоматически сохраняет историю изменений и уведомляет о важных модификациях.

ID	Название	Категория	Бренд	Артикул	Цена	Скидка (%)	Цена со скидкой	Статус	Действия
11	Сыр Российский	Сыр	Муромские Сыроварни	21512521	500,00	0	500,00	Новинка	[Редакт] [Скидка] [Отзывы] [Удалить]
12	Сыр Масдам	Сыр	вкусныйсыр	215421421	960,00	17	800,00	Рекомендуем	[Редакт] [Скидка] [Отзывы] [Удалить]
13	Свинья рулька	Свинина	СвинофермаРФ	32131294	890,00	0	890,00	Новинка	[Редакт] [Скидка] [Отзывы] [Удалить]
16	Молоко коровье	Молоко	молоко	4221421	340,00	0	340,00	Новинка	[Редакт] [Скидка] [Отзывы] [Удалить]
18	Куринные ноги	Птица	Родина	2323451	340,00	0	340,00	Новинка	[Редакт] [Скидка] [Отзывы] [Удалить]

Рисунок 7 – Таблица товаров

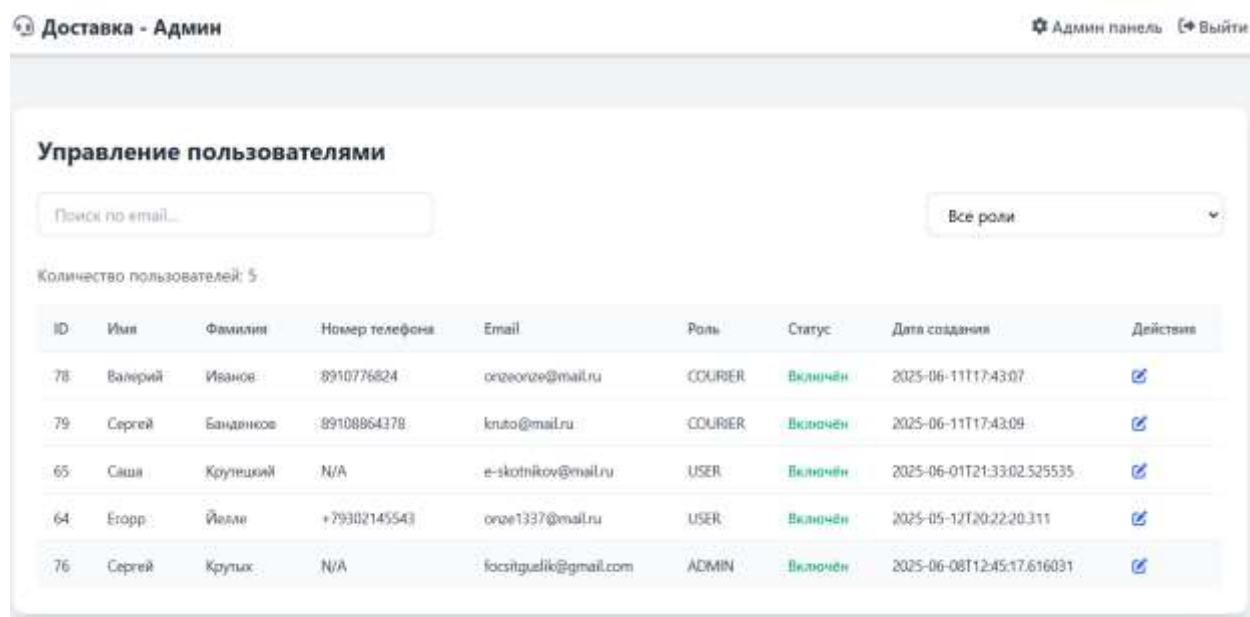
3.2.4 Управление категориями

В разделе "Категории" администратор работает с иерархической структурой категорий товаров. Интерфейс отображает дерево категорий, где можно легко просматривать связи между родительскими и дочерними категориями. При создании новой категории (Рисунок 5) администратор указывает её название, выбирает родительскую категорию и настраивает специфические поля, которые будут доступны для товаров этой категории. Это позволяет гибко настраивать структуру каталога и адаптировать её под различные типы товаров. Редактирование категорий включает возможность изменения их структуры, добавления новых подкатегорий и модификации существующих параметров.

Рисунок 8 – Добавление категории

3.2.5 Управление пользователями

Раздел "Пользователи" (Рисунок 6) предоставляет администратору доступ к полной информации о всех пользователях системы. Здесь можно просматривать профили пользователей, Администратор имеет возможность блокировать или разблокировать пользователей, а так же назначать роли.



ID	Имя	Фамилия	Номер телефона	Email	Роль	Статус	Дата создания	Действия
78	Валерий	Иванов	8910776824	onzeonze@mail.ru	COURIER	Включен	2025-06-11T17:43:07	
79	Сергей	Банденков	89108864378	knuto@mail.ru	COURIER	Включен	2025-06-11T17:43:09	
65	Саша	Крутецкий	N/A	e-skotnikov@mail.ru	USER	Включен	2025-06-01T21:33:02.525535	
64	Егор	Йелле	+79302145543	onze1337@mail.ru	USER	Включен	2025-05-12T20:22:20.311	
76	Сергей	Крутых	N/A	focsitguelik@gmail.com	ADMIN	Включен	2025-06-08T12:45:17.616031	

Рисунок 9 – Таблица с пользователями

3.2.6 Управление поддержкой

Раздел "Поддержка" представляет собой полноценную систему управления обращениями пользователей. Администратор видит все активные обращения, их статус и историю переписки. Интерфейс позволяет эффективно обрабатывать обращения, отвечать на вопросы пользователей и решать их проблемы. История всех обращений сохраняется и доступна для просмотра, что позволяет анализировать типичные проблемы и улучшать качество обслуживания.

3.2.7 Управление заказами

В разделе "Заказы" (Рисунок 7) администратор имеет полный контроль над процессом обработки заказов. Здесь отображается полный список всех заказов с детальной информацией о каждом: номер заказа, дата создания, статус, информация о клиенте, состав заказа и общая сумма. Администратор может изменять статус заказа, назначать курьеров, отслеживать процесс доставки и решать возникающие проблемы.

Рисунок 10 – Таблица с заказами

3.2.8 Управление курьерами

Раздел "Курьеры" позволяет администратору эффективно управлять службой доставки. Здесь можно просматривать информацию о всех курьерах, их текущей загрузке и эффективности работы. Администратор может добавлять новых курьеров в систему, редактировать их информацию и назначать заказы. Система автоматически отслеживает статус доставки и уведомляет о возможных задержках или проблемах.

3.3 Руководство курьера

3.2.1 Регистрация и авторизация курьера

При первом входе в систему курьер получает доступ к мобильному приложению через форму авторизации. Для входа необходимо ввести email и пароль, которые предоставляются администратором при регистрации. После успешной авторизации система генерирует JWT–токен, который используется для последующих запросов. В случае ошибки авторизации система отображает соответствующее сообщение с указанием причины (неверный email или пароль).

3.2.2 Основной интерфейс курьера

После успешного входа курьер получает доступ к основному интерфейсу приложения, который разделен на несколько ключевых разделов:

- Активные заказы – отображает список текущих заказов, назначенных курьеру;
- История заказов – содержит архив выполненных заказов;
- Профиль – содержит личную информацию и настройки;

3.2.3 Управление заказами

После успешной авторизации в системе курьер получает доступ к разделу управления заказами. В этом разделе отображается полный список всех заказов, назначенных курьеру. Система автоматически обновляет информацию в реальном времени, что позволяет курьеру всегда быть в курсе новых назначений и изменений в существующих заказах.

Для каждого заказа система предоставляет детальную информацию, включающую номер заказа, полные контактные данные клиента, точный адрес доставки, общую сумму заказа и время, когда заказ был создан. Особое внимание уделяется статусу заказа, который может быть "Назначен", "В пути",

					МИВУ.09.03.04-17.000 ВКР	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

"Доставлен" или "Отменен". Каждый статус имеет свой цветовой индикатор для быстрой визуальной идентификации.

При получении нового заказа система отправляет курьеру push-уведомление с основной информацией о заказе.

В процессе доставки курьер может отмечать различные этапы выполнения заказа. В случае возникновения проблем с доставкой (клиент не отвечает, неправильный адрес, отказ от заказа) курьер может отметить это в системе, выбрав соответствующую причину из предустановленного списка или добавив собственный комментарий.

После успешной доставки заказа курьер отмечает его как выполненный в системе. Система также предоставляет курьеру доступ к истории всех выполненных заказов. В истории сохраняется полная информация о каждом заказе, включая время доставки, адрес, сумму, статус и все комментарии, оставленные в процессе доставки. Это позволяет курьеру обращаться к информации о предыдущих заказах при необходимости.

Для удобства работы система поддерживает фильтрацию заказов по различным параметрам: по статусу, по дате, по сумме, по району доставки. Это позволяет курьеру быстро находить нужную информацию и эффективно планировать свою работу. Кроме того, система автоматически группирует заказы по районам доставки, что помогает оптимизировать маршруты и сократить время на доставку.

					МИВУ.09.03.04-17.000 ВКР	Лист
						51
Изм.	Лист	№ докум.	Подпись	Дата		

4 Тестирование системы

Тестирование веб-приложения проводилось с целью проверки работоспособности основных функций системы и их соответствия требованиям проекта. В процессе тестирования были проверены клиентская и административная части магазина, включая функционал управления товарами, заказами, пользователями и поддержкой. Тестирование осуществлялось в локальной среде разработки с использованием Spring Boot, PostgreSQL и стандартных инструментов разработки. Для тестирования использовались тестовые данные, созданные на основе типичных сценариев использования интернет-магазина.

В ходе тестирования особое внимание уделялось проверке основных бизнес-процессов, таких как регистрация и авторизация пользователей, работа с каталогом товаров, процесс оформления заказа, функционал административной панели и система поддержки пользователей. Тестирование проводилось как вручную, так и с использованием базовых инструментов тестирования API, таких как Postman, для проверки корректности работы REST-эндпоинтов.

При тестировании клиентской части были проверены все основные функции, доступные пользователям магазина. Особое внимание было уделено удобству использования интерфейса, корректности отображения информации о товарах, работе фильтров и поиска, процессу оформления заказа и функционалу личного кабинета пользователя. Все обнаруженные проблемы были зафиксированы и устранены в процессе разработки.

Тестирование административной части включало проверку функционала управления товарами, категориями, заказами и пользователями. Была проверена работа системы уведомлений, функционал обработки заказов и система поддержки пользователей. Особое внимание было уделено проверке прав доступа и безопасности системы.

					МИВУ.09.03.04-17.000 ВКР	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

4.1 Тестирование клиентской части

Клиентская часть веб–приложения была протестирована на предмет корректности работы всех основных функций и удобства использования. Особое внимание было уделено проверке различных сценариев использования, включая некорректные действия пользователей.

Процесс регистрации и авторизации был протестирован на различных сценариях. При попытке регистрации с некорректным email–адресом система корректно отображает сообщение об ошибке. Тестирование авторизации с неверными учетными данными показало, что система правильно обрабатывает такие ситуации и выводит соответствующие сообщения об ошибках. При успешной авторизации пользователь получает доступ к своему личному кабинету.

Работа с каталогом товаров была протестирована на различных сценариях. Система корректно отображает все товары, их описания, характеристики и фотографии. Фильтрация по категориям и поиск по названию работают без задержек. При попытке поиска несуществующего товара система отображает соответствующее сообщение. Тестирование показало, что система правильно обрабатывает различные параметры фильтрации и сортировки.

Процесс добавления товаров в корзину был протестирован на различных сценариях. Система корректно обрабатывает добавление товаров с указанием количества. При попытке добавить больше 5 единиц одного товара система отображает предупреждение о превышении лимита. Тестирование показало, что система правильно обновляет общую сумму заказа при изменении количества товаров.

Работа с корзиной была протестирована на различных сценариях. Система корректно обрабатывает изменение количества товаров, удаление товаров и очистку корзины. При попытке изменить количество товара на

					МИВУ.09.03.04-17.000 ВКР	Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		

некорректное значение система отображает соответствующее сообщение об ошибке.

Процесс оформления заказа был протестирован на различных сценариях. Система корректно обрабатывает указание адреса доставки, способа оплаты и подтверждение заказа. При попытке оформить заказ без указания обязательных полей система отображает соответствующие сообщения об ошибках. Тестирование показало, что система правильно обрабатывает различные способы оплаты и доставки.

Функционал личного кабинета был протестирован на различных сценариях. Система корректно отображает историю заказов, позволяет просматривать детали заказов и повторять заказы. При попытке просмотра несуществующего заказа система отображает соответствующее сообщение. Тестирование показало, что система правильно обрабатывает различные действия пользователя в личном кабинете.

Система поддержки была протестирована на различных сценариях. Система корректно обрабатывает отправку сообщений в поддержку, прикрепление файлов и просмотр истории обращений. При попытке отправить сообщение без текста система отображает соответствующее сообщение об ошибке. Тестирование показало, что система правильно обрабатывает различные типы обращений в поддержку.

В целом, тестирование клиентской части показало, что система корректно обрабатывает все основные сценарии использования и некорректные действия пользователей. Интерфейс удобен и интуитивно понятен, а система обеспечивает необходимый уровень обратной связи при возникновении ошибок.

4.2 Тестирование административной части

Управление товарами было протестировано на различных сценариях. Система корректно обрабатывает добавление новых товаров с указанием всех

					МИВУ.09.03.04-17.000 ВКР	Лист
						54
Изм.	Лист	№ докум.	Подпись	Дата		

необходимых параметров: названия, описания, цены, категории и фотографий. При попытке добавить товар без обязательных полей система отображает соответствующие сообщения об ошибках. Тестирование показало, что система правильно обрабатывает загрузку изображений различных форматов и размеров. При попытке загрузить файл неподдерживаемого формата система отображает предупреждение. Редактирование существующих товаров также работает корректно, позволяя изменять все параметры и обновлять фотографии.

Управление категориями было протестировано на различных сценариях. Система корректно обрабатывает создание новых категорий, их редактирование и удаление. При попытке удалить категорию, содержащую товары, система отображает предупреждение о невозможности удаления. Тестирование показало, что система правильно обрабатывает иерархическую структуру категорий и их отображение в каталоге.

Управление заказами было протестировано на различных сценариях. Система корректно отображает список всех заказов с возможностью фильтрации по статусу. Тестирование показало, что система правильно обрабатывает назначение курьеров к заказам и отслеживание статуса доставки.

Управление пользователями было протестировано на различных сценариях. Тестирование показало, что система правильно обрабатывает различные действия с пользователями, включая блокировку, разблокировку и смену их ролей.

Система поддержки была протестирована на различных сценариях. Система корректно отображает все обращения пользователей с возможностью фильтрации по статусу и email. Тестирование показало, что система правильно обрабатывает ответы на обращения, прикрепление файлов и изменение статуса обращений.

					МИВУ.09.03.04-17.000 ВКР	Лист
						55
Изм.	Лист	№ докум.	Подпись	Дата		

Управление курьерами было протестировано на различных сценариях. Система корректно обрабатывает добавление новых курьеров, их редактирование и удаление. Тестирование показало, что система правильно обрабатывает назначение заказов курьерам и отслеживание их статуса.

В целом, тестирование административной части показало, что система корректно обрабатывает все основные сценарии работы администраторов и некорректные действия. Интерфейс административной панели удобен и интуитивно понятен, а система обеспечивает необходимый уровень обратной связи при возникновении ошибок. Все функции управления магазином работают стабильно и обеспечивают эффективное администрирование.

4.3 Тестирование мобильного приложения для курьеров

Мобильное приложение для курьеров было протестировано на различных устройствах и операционных системах. Особое внимание было уделено проверке удобства использования, производительности и корректности работы всех функций, необходимых для эффективной работы курьеров. Процесс авторизации был протестирован на различных сценариях. Приложение корректно обрабатывает вход с использованием email и пароля. При попытке входа с некорректными данными система отображает понятные сообщения об ошибках. Работа с активными заказами была протестирована на различных сценариях. Приложение корректно отображает список назначенных заказов с детальной информацией: адрес доставки, контакты клиента, состав заказа и сумма. При получении нового заказа система отправляет push-уведомление и обновляет список. Тестирование показало, что система правильно обрабатывает различные статусы заказов и их изменения. При попытке изменить статус заказа на некорректный система отображает соответствующее сообщение об ошибке.

					МИВУ.09.03.04-17.000 ВКР	Лист
						56
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В ходе выполнения выпускной квалификационной работы была разработана и реализована система онлайн-заказа и доставки продуктов питания.

Проектирование системы включало детальное моделирование предметной области, разработку диаграмм вариантов использования и проектирование архитектуры приложения. В результате была создана многофункциональная платформа, обеспечивающая удобный процесс покупок для клиентов и эффективное управление магазином для администраторов.

Реализация системы выполнена с использованием современных технологий: Spring Boot в качестве backend-фреймворка и PostgreSQL в качестве системы управления базами данных. Особое внимание было уделено вопросам безопасности и соответствия законодательным требованиям, включая защиту персональных данных и реализацию механизмов, соответствующих нормам Гражданского кодекса РФ и закона "О защите прав потребителей".

Разработанное решение обладает значительным потенциалом для дальнейшего развития. Перспективными направлениями совершенствования системы являются внедрение рекомендательных алгоритмов на основе машинного обучения, разработка полноценного мобильного приложения, развитие мобильного курьерского приложения и интеграция с современными сервисами логистики.

Таким образом, поставленные в работе задачи успешно решены, а разработанная система представляет собой полноценное конкурентноспособное решение для рынка онлайн-доставки продуктов питания, сочетающее удобство использования, функциональность и соответствие актуальным требованиям законодательства.

					МИВУ.09.03.04-17.000 ВКР	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

Список литературы

1) Гражданский кодекс Российской Федерации (часть вторая) от 26.01.1996 № 14-ФЗ (ред. от 01.07.2024) // Собрание законодательства РФ. — 1996. — № 5. — Ст. 410. — URL: http://www.consultant.ru/document/cons_doc_LAW_9027/ (дата обращения: 15.05.2025).

2) О защите прав потребителей: Федеральный закон от 07.02.1992 № 2300-1 (ред. от 08.12.2023) // Собрание законодательства РФ. — 1992. — № 15. — Ст. 766. — URL: http://www.consultant.ru/document/cons_doc_LAW_305/ (дата обращения: 15.05.2025).

3) О персональных данных: Федеральный закон от 27.07.2006 № 152-ФЗ (ред. от 02.07.2024) // Собрание законодательства РФ. — 2006. — № 31 (ч. I). — Ст. 3451. — URL: http://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 15.05.2025).

4) Условия использования API Яндекс.Карт [Электронный ресурс]. — URL: https://yandex.ru/legal/maps_api/ (дата обращения: 15.05.2025).

					МИВУ.09.03.04-17.000 ВКР	Лист
						58
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А

(обязательное)

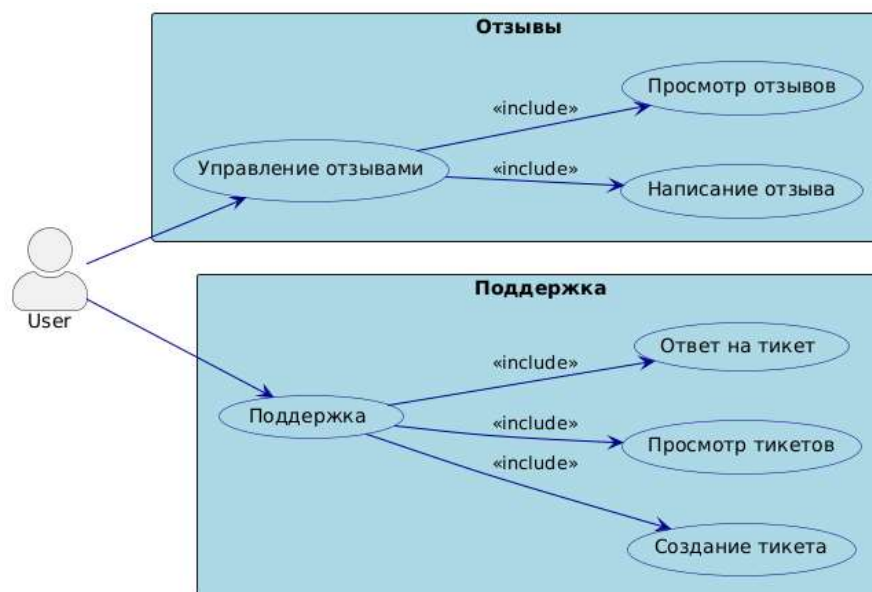


Рисунок А.1– Диаграмма вариантов использования пользователя (Отзывы, Поддержка)

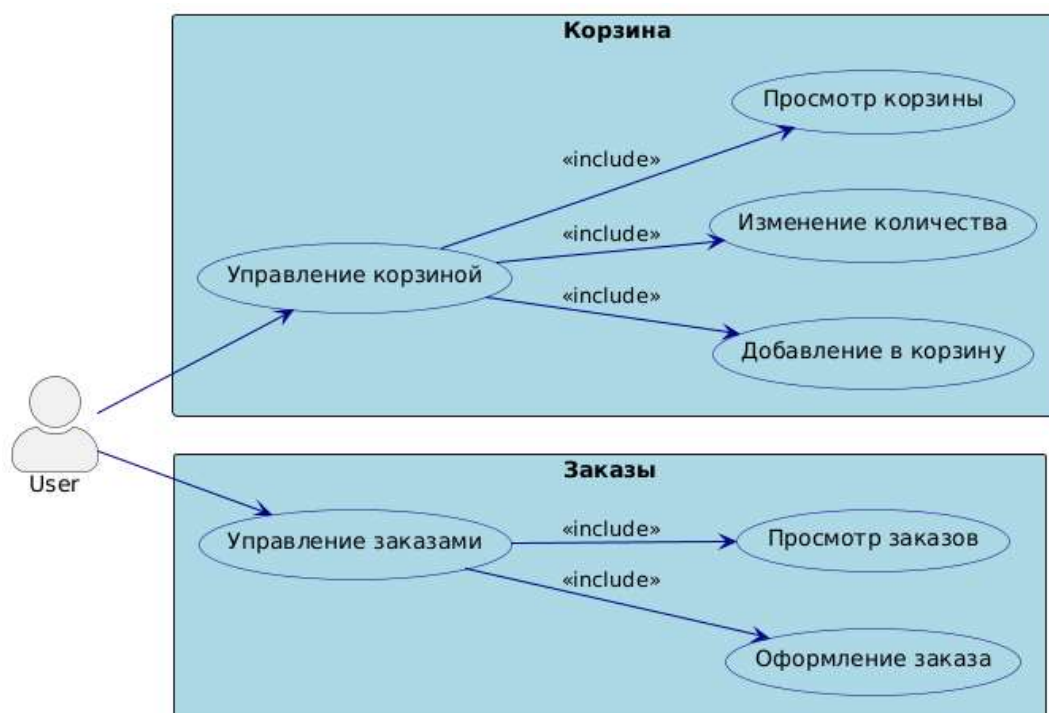


Рисунок А.2– Диаграмма вариантов использования пользователя (Корзина, Заказы)

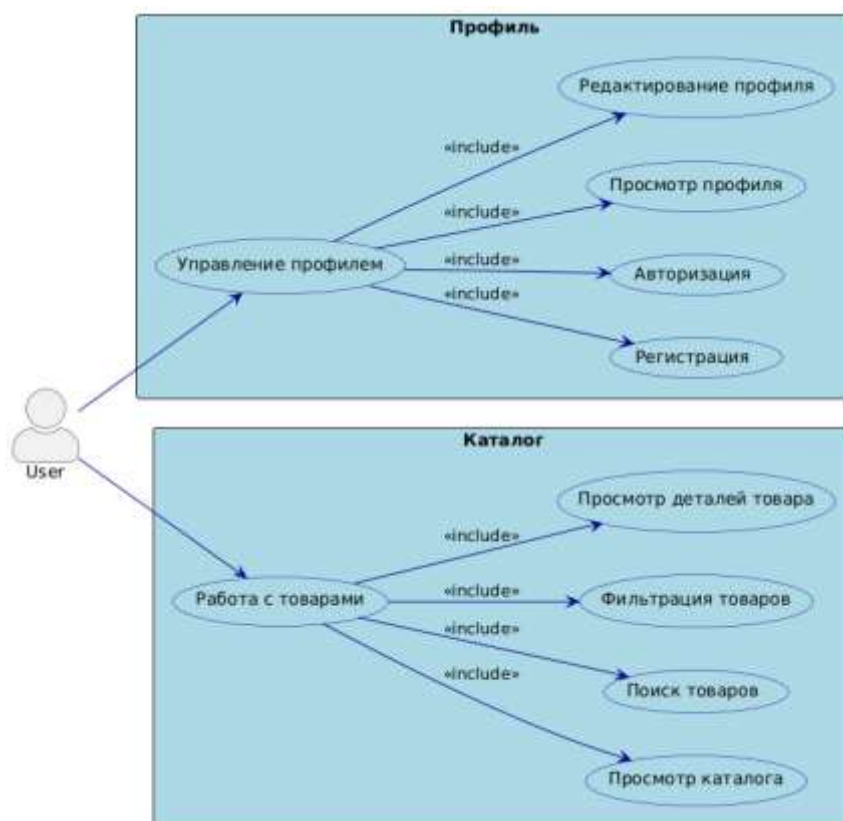


Рисунок А.3– Диаграмма вариантов использования пользователя (Профиль, Каталог)



Рисунок А.4– Диаграмма вариантов использования администратора (Управление контентом)

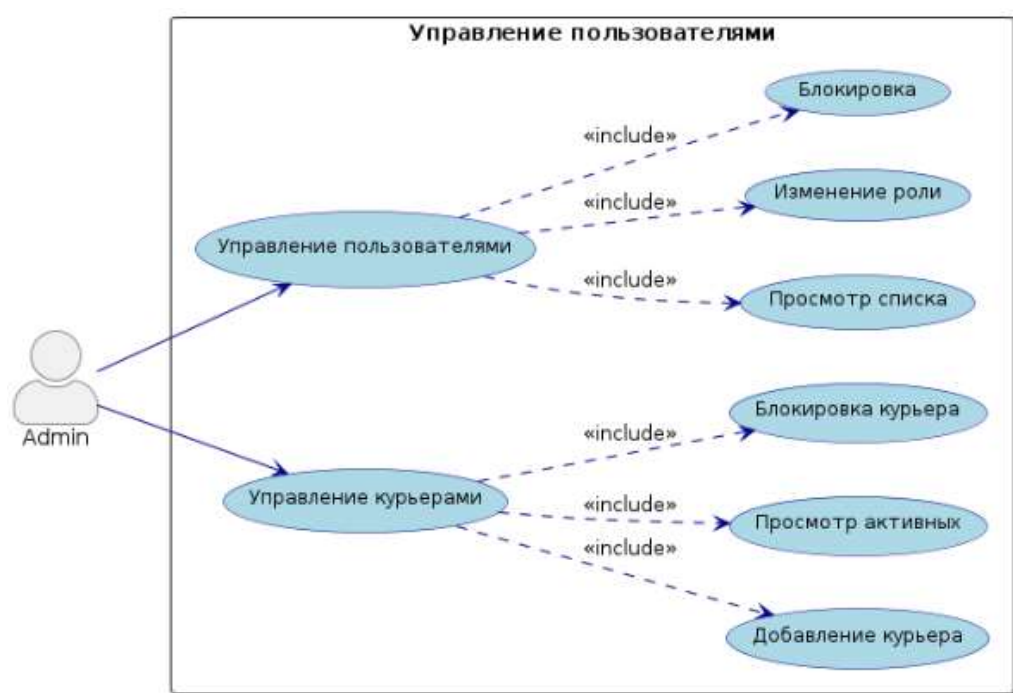


Рисунок А.5– Диаграмма вариантов использования администратора
(Управление пользователями)

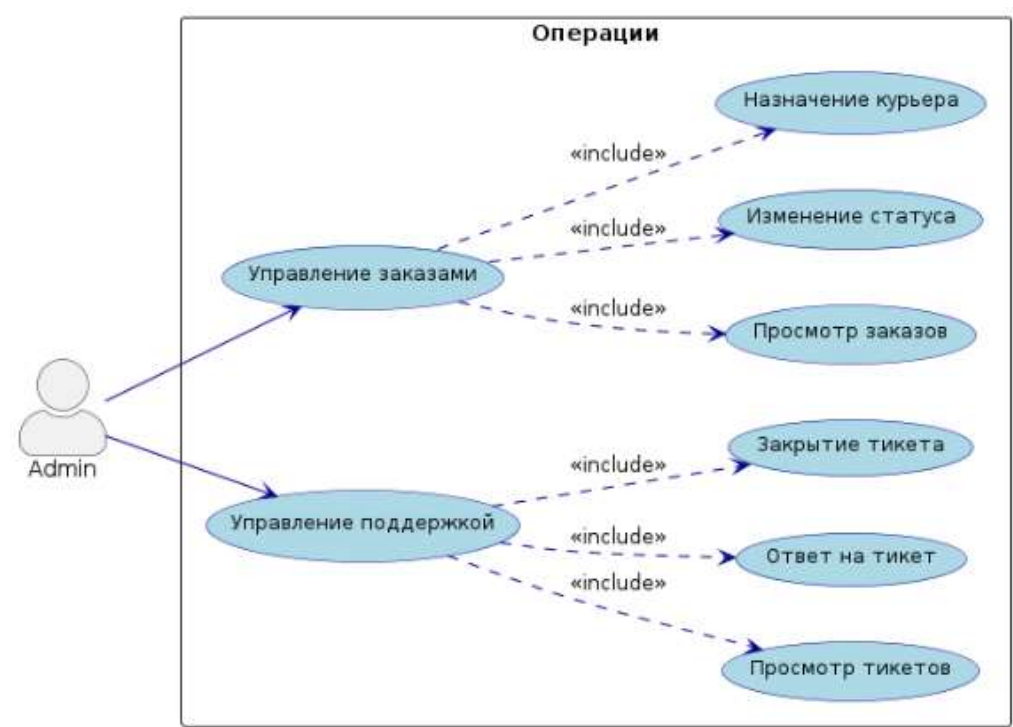


Рисунок А.6– Диаграмма вариантов использования администратора
(Операции)

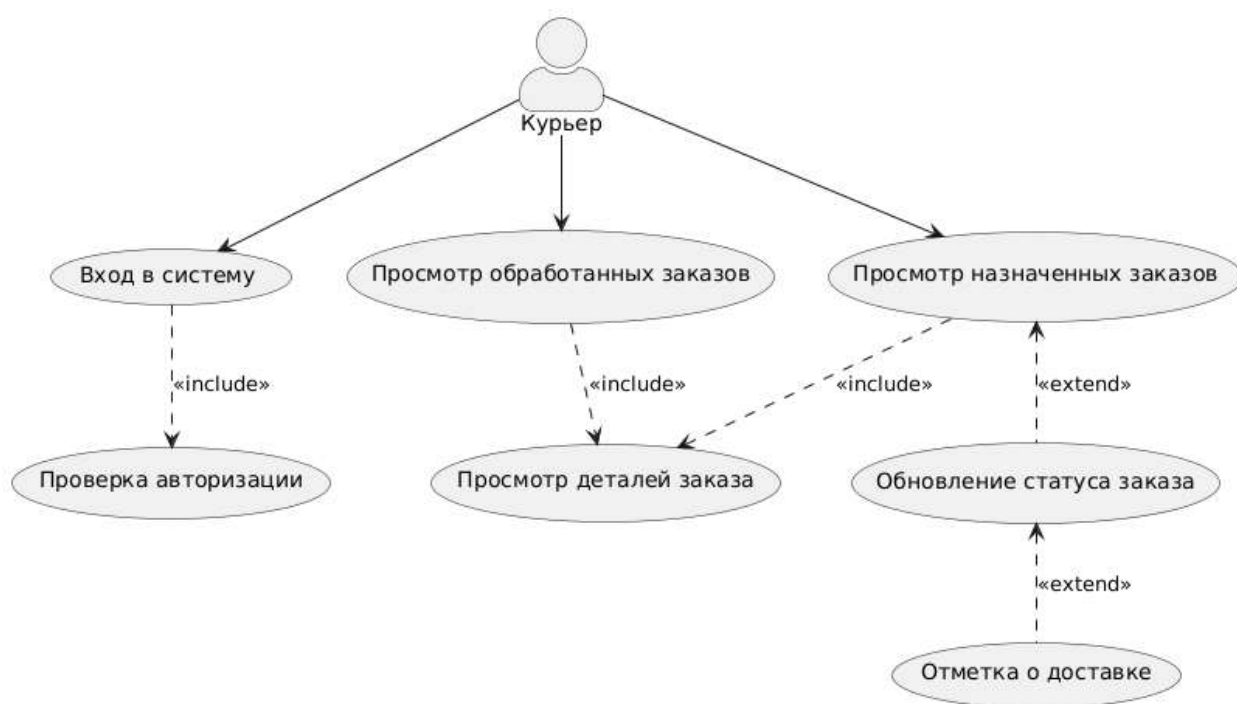


Рисунок А.7– Диаграмма вариантов использования курьера

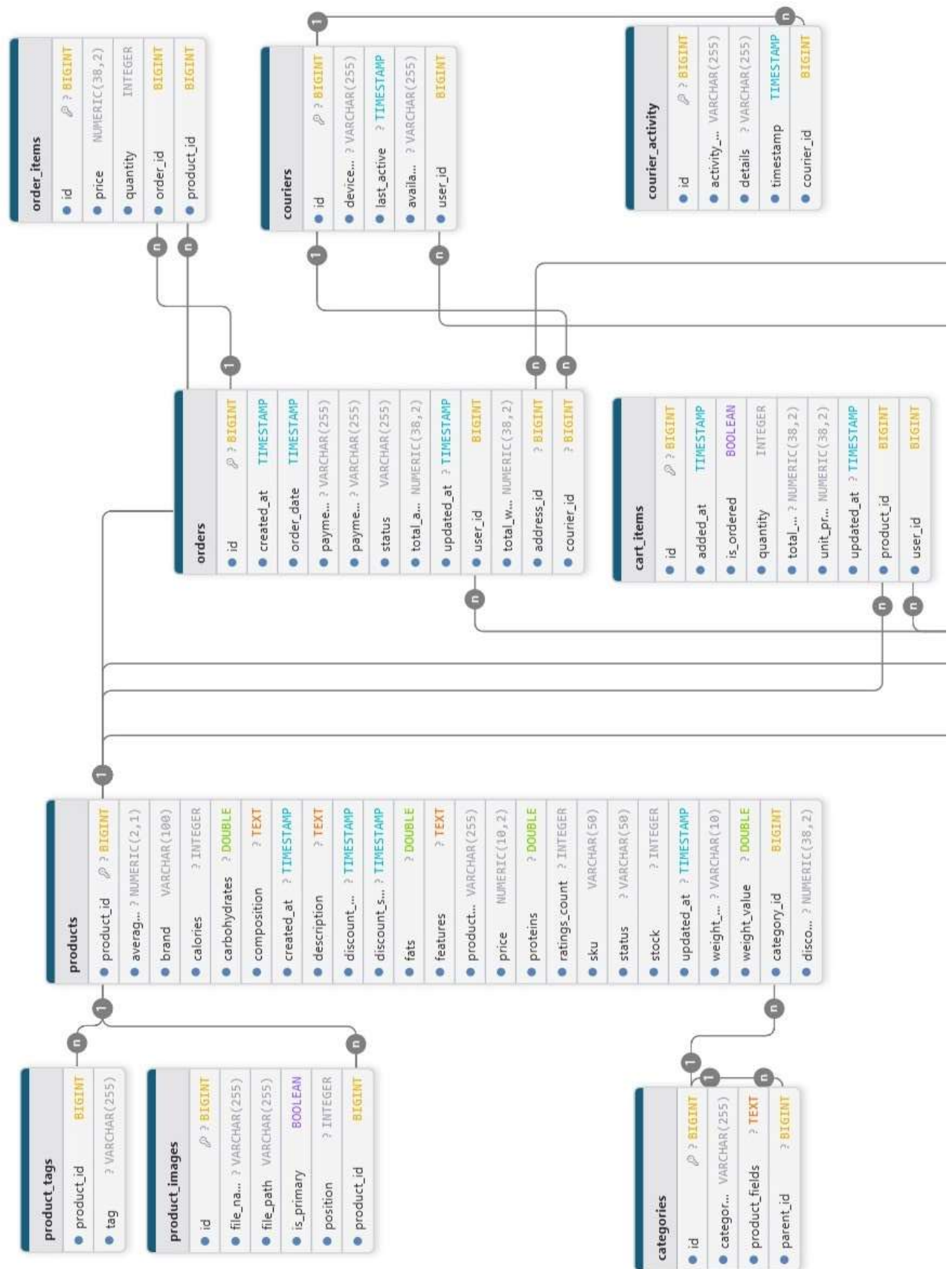


Рисунок А.8 – Физическая модель базы данных (часть 1)

Приложение Б

Скриншоты окон программы (справочное)

Вход в аккаунт

Неверный email или пароль

Email адрес

Пароль

Войти

Ещё нет аккаунта? [Зарегистрируйтесь](#)

Рисунок Б.1 – Страница с заказами

Активные заказы

Заказ #15

14.06.2025 09:03

В обработке

Итого: ₽ 4560,00

Последние завершённые заказы

У вас нет завершённых заказов

Рисунок Б.2 – Страница с заказами

					МИВУ.09.03.04-17.000 ВКР	Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		

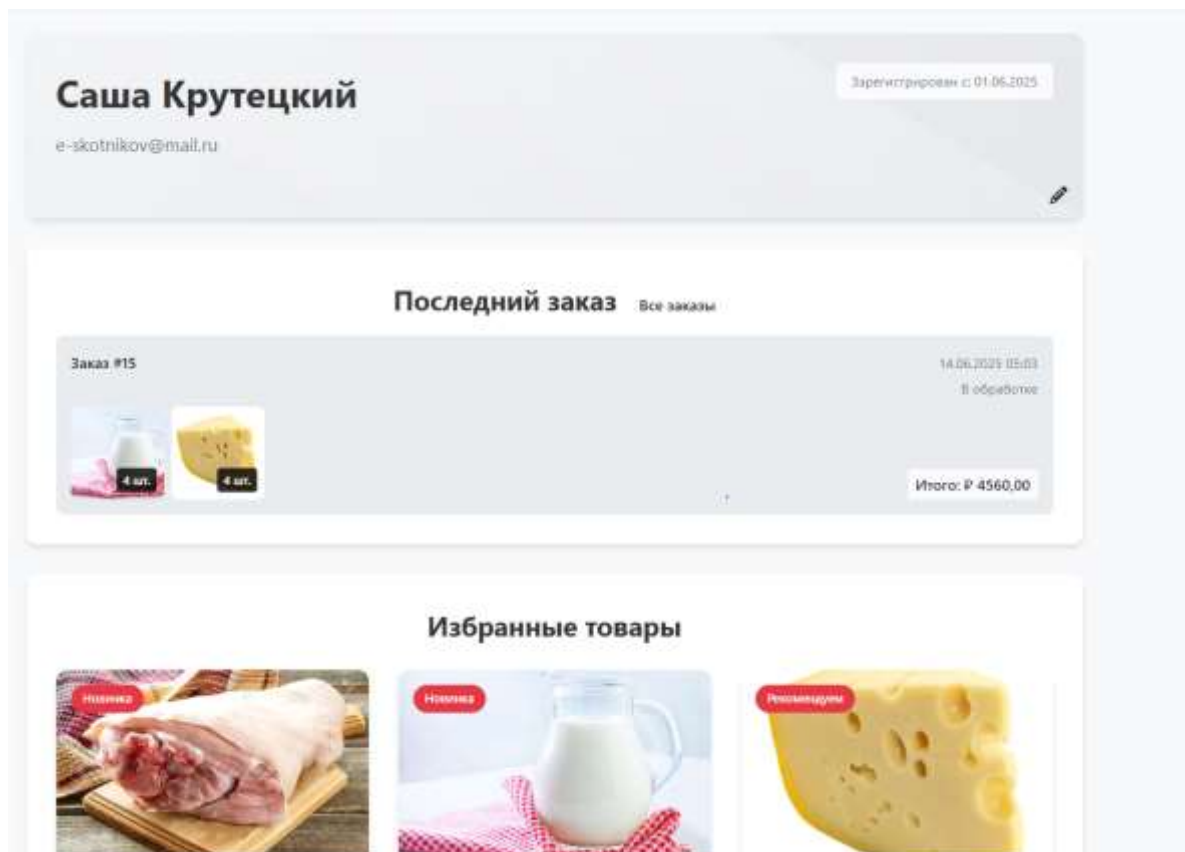


Рисунок Б.3 – Страница пользователя

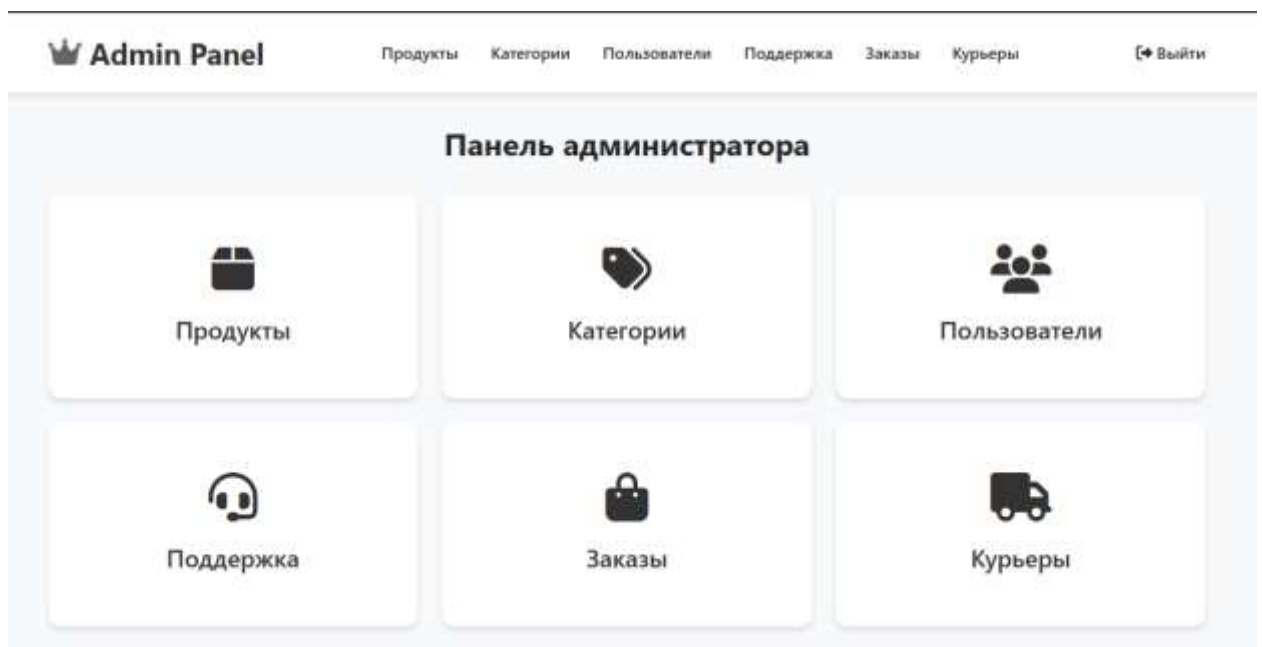


Рисунок Б.4 – Панель администратора

					МИВУ.09.03.04-17.000 ВКР	Лист
						66
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение В

Код программы

(справочное)

Код разработанного приложения доступен на веб-сервисе для хостинга IT-проектов Github по адресу: <https://github.com/Onzeboy/vkr>

					МИВУ.09.03.04-17.000 ВКР	Лист
						67
Изм.	Лист	№ докум.	Подпись	Дата		