

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.04.04 - Программная инженерия
Направленность (профиль) образовательной программы Управление разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«__» _____ 2025 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Реализация системы управления продажами для цветочных и зоомагазинов с использованием Java

Исполнитель Студент группы 3105-ом2	_____	Сунь Чжаожуй
	(подпись, дата)	
Руководитель доцент, канд. техн. наук	_____	Л.В. Никифорова
	(подпись, дата)	
Руководитель научного содержания программы магистратуры профессор, доктор техн. наук	_____	И.Е. Еремин
	(подпись, дата)	
Нормоконтроль инженер кафедры	_____	В.Н. Адаменко
	(подпись, дата)	
Рецензент директор ЦЦТиТО	_____	А.А. Тодосейчук
	(подпись, дата)	

Благовещенск, 2025

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем

УТВЕЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
« ____ » _____ 2025 г.

З А Д А Н И Е

К магистерской диссертации студента группы 3105-ом2 Сунь Чжаожуй

1. Тема магистерской диссертации: Реализация системы управления продажами для цветочных и зоомагазинов с использованием Java

(Утверждено приказом от 06.03.2025 № 609-уч)

2. Срок сдачи студентом законченной работы (проекта): 10.06.2025

3. Исходные данные к магистерской диссертации: документация разработчиков, интернет-ресурсы, учебная литература

4. Содержание магистерской диссертации (перечень подлежащих разработке вопросов): проектирование алгоритма решения задачи, работа с базой данных

5. Перечень материалов приложения (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.): нет

6. Рецензент магистерской диссертации: Тодосейчук А.А., директор ЦЦТиТО

7. Дата выдачи задания 17.01.25

8. Руководитель выпускной квалификационной работы : _____

Л.В. Никифорова, доцент, канд. техн. наук

(фамилия, имя, отчество, должность, уч.степень, уч.звание)

Заявление принял к исполнению _____

РЕФЕРАТ

Магистерская диссертация содержит 82 страниц, 27 рисунков, 43 источника, 18 таблиц

ЦВЕТОЧНЫЙ И ЗООМАГАЗИН, JAVA, ИНФОРМАЦИОННАЯ СИСТЕМА, ПРОДАЖИ, АВТОМАТИЗАЦИЯ

Целью данной выпускной квалификационной работы является разработка и реализация информационной системы управления продажами для специализированного магазина, совмещающего торговлю цветами и зоотоварами, с использованием языка программирования Java.

Разрабатываемое программное обеспечение обеспечивает удобный интерфейс для сотрудников и администраторов магазина, а также предоставляет функциональные модули для учёта товаров, фиксации продаж, анализа остатков и формирования отчетов.

Процесс создания программного продукта включал следующие этапы:

- анализ предметной области и сбор требований к системе;
- проектирование архитектуры программного обеспечения;
- реализация функциональных модулей с использованием Java;
- моделирование бизнес-процессов с помощью UML-диаграмм;
- тестирование и отладка приложения;
- оформление пользовательской документации.

Разработанная система управления продажами демонстрирует соответствие практическим потребностям небольших розничных торговых точек и может быть адаптирована под конкретные условия бизнеса. Проект обладает прикладной значимостью и может быть использован как основа для цифровизации деятельности цветочных и зоомагазинов.

СОДЕРЖАНИЕ

Введение	6
1 Предметная область исследования	8
1.1 Справочная информация по теме	8
1.2 Анализ современного состояния проблемы исследования	11
1.2.1 Международные исследования	12
1.3 Цель и задачи	14
1.4 Описание методов и инструментов разработки	15
1.4.1 Методы разработки	15
1.4.2 Средства разработки	16
1.4.3 Методы разработки информационных систем	18
2 Алгоритмическое и программное обеспечение решения задачи продаж	24
2.1 Анализ осуществимости	24
2.1.1 Экономическое обоснование	24
2.1.2 Эксплуатационная осуществимость	24
2.1.3 Техническая осуществимость	26
2.1.4 Схема системы	27
2.2 Анализ потребностей	29
2.2.1 Функциональные требования	29
2.2.2 Требования к характеристикам	32
2.3 Обзор возможностей профильного программного обеспечения	35
2.3.1 Платформы и фреймворки	36
2.3.2 Среда и инструменты разработки	37
2.3.3 Архитектурный стиль	42
3 Программная реализация предлагаемого алгоритма решения задачи продаж	45
3.1 Основные этапы практической разработки программного продукта	45
3.1.1 Проектирование архитектуры системы	45
3.1.2 Разработка структуры системы	51

3.2. Примеры фактического тестирования программного продукта	58
3.2.1 Структура пользовательского интерфейса	58
3.2.2 Тестирование пользовательского интерфейса	74
Заключение	80
Библиографический список	81

ВВЕДЕНИЕ

В условиях стремительного развития цифровых технологий и усиления конкуренции в розничной торговле автоматизация процессов управления продажами становится ключевым фактором успешного функционирования предприятий. Особенно это актуально для специализированных магазинов, таких как цветочные и зоомагазины, деятельность которых характеризуется широким ассортиментом, высокой оборачиваемостью товара, сезонными колебаниями спроса и необходимостью индивидуального подхода к каждому клиенту.

Современные тенденции в области цифровизации бизнеса ориентированы на широкое внедрение информационных систем, способных не только автоматизировать рутинные операции, но и предоставлять инструменты для аналитики, прогнозирования спроса, формирования клиентской лояльности. Учитывая это, разработка специализированного программного обеспечения для малого розничного бизнеса приобретает особую актуальность.

Актуальность темы дипломной работы обусловлена необходимостью повышения эффективности бизнес-процессов и качества обслуживания покупателей путём внедрения современных информационных систем. Использование языка программирования Java как технологической основы разработки позволяет создать надёжное, масштабируемое и кроссплатформенное решение, ориентированное на нужды малого и среднего бизнеса. Java обладает развитой экосистемой, поддержкой большого количества фреймворков и библиотек, что способствует ускорению разработки и повышению гибкости программных решений.

Цель дипломной работы – разработка и реализация системы управления продажами для цветочных и зоомагазинов с использованием языка программирования Java.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области и выделить ключевые бизнес-процессы;

- определить функциональные и архитектурные требования к системе;
- спроектировать структуру базы данных и пользовательский интерфейс;
- разработать программное обеспечение с использованием Java и сопутствующих технологий;
- выполнить тестирование и оценку эффективности разработанной системы.

Объектом исследования является процесс управления продажами в специализированных розничных магазинах – цветочных и зоомагазинах.

Предмет исследования – методы и средства программной реализации систем автоматизации управления продажами.

Разработка подобных систем требует учёта множества факторов: удобства интерфейса для конечного пользователя, скорости обработки операций, безопасности хранения данных, а также возможности интеграции с внешними сервисами, такими как платёжные шлюзы, службы доставки или CRM-платформы.

Практическая значимость работы заключается в возможности применения созданной системы в деятельности реальных торговых предприятий с целью оптимизации процессов учёта, повышения скорости и точности оформления заказов, улучшения контроля за товарными остатками и анализа продаж. Внедрение разработанного решения может способствовать цифровой трансформации торговых процессов и росту конкурентоспособности малого бизнеса.

1 ПРЕДМЕТНАЯ ОБЛАСТЬ ИССЛЕДОВАНИЯ

1.1 Справочная информация по теме

Магазин цветов и птиц – это розничный магазин, специализирующийся на продаже цветов, растений, птиц и сопутствующих товаров. С повышением уровня жизни людей и изменением эстетических представлений рынок цветов и птиц постепенно расширяется, а потребительский спрос на цветы и растения растет. Современные магазины цветов и птиц должны обеспечивать не только разнообразный выбор товаров, но и отличное обслуживание клиентов, чтобы удовлетворить индивидуальные потребности покупателей.

Магазины цветов и птиц объединяют два ключевых направления: флористику и продажу декоративных птиц. Это специализированные точки, где покупатели находят не только сами товары, но и идеи для украшения интерьера, обустройства сада или создания уютной домашней атмосферы. Магазины этого типа часто включают товары для ухода за растениями и птицами, аксессуары для содержания животных и уникальные дизайнерские композиции.

Спрос на услуги таких магазинов растет в силу нескольких факторов. Во-первых, увеличивается популярность экологичного образа жизни и стремление горожан добавить природные элементы в свой быт. Во-вторых, цветы и птицы часто используются как символы радости и гармонии, что делает их востребованными в качестве подарков.

Однако традиционные методы управления этими магазинами часто сталкиваются с рядом проблем. В частности, обработка данных вручную может занимать значительное время, а ошибки при подсчете или управлении запасами приводят к финансовым потерям. Сложности также возникают при работе с сезонными товарами, которые требуют особого подхода к учету и хранению.

Управление сезонными цветами, такими как тюльпаны или розы на праздники, может быть неэффективным без учета текущих трендов спроса. Отсутствие точного анализа данных может привести к как дефициту, так и перепроизводству.

Современные системы автоматизации продаж обеспечивают значительное упрощение этих процессов. Внедрение технологий позволяет автоматизировать учет, обеспечить качественное обслуживание клиентов и повысить уровень взаимодействия с покупателями.

В традиционных магазинах цветов и птиц управление продажами в значительной степени зависит от ручного учета или простых электронных таблиц, и здесь существует несколько основных проблем:

Сложность управления запасами: разнообразие и сезонность цветочной и птичьей продукции усложняют управление запасами. Традиционный ручной метод учета чреват такими проблемами, как запаздывание данных, ошибки и информационная асимметрия.

Громоздкая обработка данных о продажах: учет продаж в основном ведется в виде бумажных бухгалтерских книг или простых электронных таблиц, а обработка данных недостаточно эффективна для своевременного составления отчетов о продажах и анализа данных.

Ограниченный уровень обслуживания клиентов: отсутствие систематического управления информацией о клиентах и неспособность эффективно отслеживать историю покупок и предпочтения клиентов приводят к низкой степени персонализации обслуживания клиентов и маркетинговых мероприятий.

Недостаточные возможности анализа данных: в традиционном режиме анализ данных обычно основывается на ручной статистике и анализе, что неэффективно и чревато ошибками, влияющими на принятие магазином бизнес-решений и выработку стратегии.

Для того чтобы справиться с вышеперечисленными проблемами, особенно важно преобразовать информационные технологии магазинов цветов и птиц. Внедрение системы продаж может значительно улучшить недостатки традиционной модели продаж и принести пользу в следующих областях:

Автоматизированное управление запасами: система автоматически отслеживает изменения запасов, обновляет информацию о запасах в режиме реального времени, сокращает количество ошибок при ручных операциях, обеспечивает

точность и доступность данных о запасах в режиме реального времени.

Эффективный учет продаж – система автоматически регистрирует каждую транзакцию и формирует подробные данные о продажах и отчеты, облегчая анализ и архивирование данных о продажах.

Управление информацией о клиентах – централизованное управление информацией о клиентах, отслеживание истории их покупок, предоставление персонализированных услуг и маркетинговых рекомендаций для повышения удовлетворенности и лояльности клиентов.

Анализ данных и поддержка принятия решений – система предоставляет мощные функции анализа данных и генерирует различные виды бизнес-отчетов, помогая магазинам понять тенденции продаж и оптимизировать бизнес-стратегии и управление запасами.

В соответствии с вышеизложенным, система продаж для магазинов цветов и птиц должна обладать следующими основными функциями.

Управление запасами:

- обновление состояния запасов в режиме реального времени;
- учет поступления и отсутствия товаров на складе;
- автоматическое формирование отчетов о запасах.

Управление продажами:

- обработка операций по продажам;
- составление отчетов о продажах;
- анализ тенденций и показателей продаж.

Управление клиентами:

- запись и управление информацией о клиентах;
- отслеживание истории их покупок;
- реализация персонализированных маркетинговых стратегий.

Анализ данных:

- обобщение данных о продажах;
- создание отчетов по бизнес-анализу для поддержки принятия решений.

Генерация отчетов:

- автоматическая генерация;
- экспорт всех видов бизнес-отчетов для визуального отображения данных.

С непрерывным прогрессом технологий, тенденция развития системы продаж магазина цветов и птиц включает в себя:

Интеллектуальность – внедрение технологий искусственного интеллекта, таких как поиск данных и прогнозный анализ, для дальнейшего повышения уровня интеллектуальности системы.

Мобилизация – разработка мобильных приложений, позволяющих магазинам управлять своим бизнесом в любое время и в любом месте и повышающих гибкость работы.

Интеграция – взаимодействие с другими бизнес-системами (например, системами управления цепочками поставок, системами управления взаимоотношениями с клиентами) для обеспечения обмена информацией и делового сотрудничества в масштабах всей сети.

1.2 Анализ современного состояния проблемы исследования

В Китае многие магазины цветов и птиц начали внедрять системы продаж для повышения эффективности управления [36]. Функции системы в основном включают управление запасами, учет продаж, управление клиентами и анализ данных. Однако многие малые и средние магазины цветов и птиц все еще используют простые электронные таблицы или традиционные ручные записи, с разной степенью проникновения и применения системы.

С развитием электронной коммерции и цифровой трансформации магазинов цветов и птиц уделяют все больше внимания сочетанию онлайн и офлайн. Система не только поддерживает управление магазином, но и интегрирует платформу онлайн-продаж для достижения омни-канальной работы [32]. Высококлассные магазины цветов и птиц в крупных городах обычно используют более продвинутые системы для поддержки интеллектуального управления запасами и персонализированного обслуживания клиентов.

Отечественные магазины цветов и птиц сталкиваются с такими проблемами:

- стоимость системы;
- сложность эксплуатации;
- техническая поддержка при внедрении систем продаж.

По мере развития технологий и увеличения спроса на рынке появляется все больше решений и поставщиков услуг, что способствует росту популярности системы и улучшению ее функциональности.

1.2.1 Международные исследования

В развитых странах, таких как США, Европа и Япония, системы продаж в магазинах цветов и птиц уже относительно развиты. На международном уровне многие магазины цветов и птиц используют системы, объединяющие передовые функции анализа данных и управления взаимоотношениями с клиентами (CRM) для поддержки межрегиональных и многоканальных операций.

Технологии искусственного интеллекта и больших данных широко внедряются в системы продаж магазинов цветов и птиц на международном рынке. Система используется не только для ежедневной работы и управления, но и для точного прогнозирования рынка и персонализированных рекомендаций. Технология Интернета вещей (IoT) также постепенно применяется для мониторинга и управления товарными запасами и условиями окружающей среды в режиме реального времени.

Магазины цветов и птиц на международном рынке все больше внимания уделяют бесшовной интеграции с платформами электронной коммерции для поддержки услуг онлайн-заказа и бронирования. Между тем, экологичное и устойчивое развитие становится тенденцией рынка, и система интегрирует соответствующие функции, такие как отслеживание экологических данных и управление энергосбережением.

В России и странах СНГ цифровизация магазинов цветов и птиц развивается неравномерно. Крупные сети уже используют специализированное ПО, тогда как малый бизнес ограничен в ресурсах. Проблемы внедрения, такие как высокая стоимость программных продуктов и сложность их адаптации под локальные нужды, остаются актуальными.

Однако есть положительные примеры. Так, внедрение CRM-систем в крупных флористических магазинах позволило увеличить повторные продажи на 25%, благодаря анализу предпочтений клиентов и персонализированным предложениям. Оценка эффективности внедрения CRM-системы показана на рисунке 1.



Рисунок 1 – Оценка эффективности внедрения CRM-системы

Преимущества внедрения CRM Приведенная классификация является достаточно наглядной, однако она не учитывает эффект снижения рисков. Эффективность внедрения CRM выражается в трех условных категориях: прямые эффекты; косвенные экономические эффекты; снижение рисков.

Кроме того, многие компании переходят к использованию омниканальных решений – интеграция интернет-магазинов с физическими точками продажи становится трендом.

В международной практике использование систем на основе искусственного интеллекта для управления продажами становится стандартом. Например, в Японии магазины цветов внедряют системы, анализирующие погодные условия и исторические данные для прогноза спроса. В США компании, занимающиеся продажей птиц, используют IoT для мониторинга условий содержания животных

прямо в торговом зале.

В Европе уделяется большое внимание экологическим стандартам, и многие системы интегрируют контроль за экологической устойчивостью бизнеса.

В Азии приоритетом является масштабируемость и мобильность систем, позволяющих управлять несколькими магазинами с одного устройства.

В России акцент смещен на снижение затрат и интеграцию базовых функций.

1.3 Цель и задачи

При создании магазина цветов и птиц следует ориентироваться на следующие виды целей:

- повышение эффективности;
- оптимизация работы с клиентами;
- анализ данных;
- расширенные возможности управления.

Повышение эффективности подразумевает автоматизацию управления запасами, обработку заказов и учет продаж для сокращения ошибок и временных затрат на ручные операции.

Оптимизация работы с клиентами обозначает обеспечение легкого доступа к покупкам, управление информацией о клиентах и историей заказов для лучшего обслуживания и удовлетворения потребностей клиентов.

Анализ данных – это сбор и анализ данных о продажах, чтобы помочь магазинам понять тенденции продаж, предпочтения покупателей и разработать более эффективные маркетинговые стратегии.

Расширенные возможности управления представляет мониторинг товарных запасов в режиме реального времени, финансовую отчетность и статистику продаж. Это помогает магазинам лучше распределять ресурсы и принимать решения.

Таким образом, целью исследования является разработка автоматизированной системы продаж, которая позволит:

- улучшить управление запасами;

- повысить качество обслуживания;
- интегрировать данные для стратегического планирования.

Для этого стоит решить следующие задачи:

- разработка интерфейса для работы с данными о запасах, продажах и клиентах;
- создание модулей для аналитики продаж и формирования отчетности;
- интеграция мобильного приложения для управления системой на базе Java;
- внедрение инструментов, обеспечивающих прогнозирование спроса.

1.4 Описание методов и инструментов разработки

1.4.1 Методы разработки

Методология разработки делится на следующие шаги:

- анализ требований;
- проектирование системы;
- разработка прототипа;
- разработка системы;
- тестирование и отладка;
- развертывание и сопровождение.

Целью анализа требований является понять специфические требования магазина, включая функциональные требования, требования пользователей и требования к производительности системы.

Методами является сбор требований путем общения с заказчиками, анкетирование, наблюдение на месте.

Целью проектирования системы является планирование общей архитектуры системы и проектирование детальных функциональных модулей.

Методами является использование UML (Unified Modelling Language) для проектирования и создания диаграмм архитектуры системы, диаграмм модели данных, диаграмм вариантов использования [25].

Целью разработки прототипа является быстрое создание прототипов системы для проверки обоснованности требований и дизайна.

Методами является использование инструментов прототипирования для создания прототипов пользовательского интерфейса, проведения пользовательского тестирования и корректировки обратной связи.

Целью разработки системы является разработка кода и внедрение системы в соответствии с проектной документацией.

Методами является итеративная разработка и тестирование с использованием методов Agile-разработки.

Целью тестирования и отладки является обеспечение корректности и стабильности функциональности системы.

Методами является проведение модульного тестирования, интеграционного тестирования, системного тестирования и приемочного тестирования для устранения найденных ошибок и проблем.

Целью развертывания и сопровождения является развертывание системы в производственной среде и выполнение текущего обслуживания и обновлений.

Методами является использование автоматизированных средств развертывания для развертывания системы, регулярного обслуживания и обновления системы, а также обработки отзывов пользователей и новых требований [17].

1.4.2 Средства разработки

Языки программирования и фреймворки:

- широко используемый для разработки приложений корпоративного уровня и отличающийся высокой стабильностью язык Java;
- простой в разработке и сопровождении, подходящий для обработки данных и быстрой разработки Python;
- используемый для фронтенд-разработки, с такими фреймворками, как React, Angular или Vue.js, для динамической разработки пользовательского интерфейса JavaScript.

Система управления базами данных:

- реляционная база данных с открытым исходным кодом, подходящая для малых и средних приложений MySQL;
- реляционная база данных с открытым исходным кодом, поддерживающая

сложные запросы и целостность данных PostgreSQL;

- база данных NoSQL с открытым исходным кодом, подходящая для обработки неструктурированных данных и высоких требований к параллельности MongoDB.

Фреймворки и платформы для разработки:

- упрощающий разработку Java-приложений, поддерживающий быстрое создание приложений корпоративного класса Spring Boot;

- эффективный веб-фреймворк для быстрой разработки и создания прототипов Django;

- современный PHP-фреймворк для поддержки быстрого создания и сопровождения веб-приложений Laravel.

Интегрированная среда разработки:

- мощная IDE для разработки на Java, поддерживающая множество языков и фреймворков IntelliJ IDEA;

- IDE, предназначенная для разработки на языке Python PyCharm;

- легкий редактор кода, поддерживающий множество языков программирования и плагинов Visual Studio Code.

Система контроля версий:

- распределенная система контроля версий, поддерживающая совместную работу и версионирование Git;

- сервис онлайн-хостинга, поддерживающий размещение кода, совместную работу и интеграцию CD GitHub.

Инструменты управления проектами:

- для отслеживания проблем, задач и хода проекта, поддерживает agile-разработку Jira;

- визуальный инструмент управления проектами для управления задачами и рабочими процессами Trello.

Инструменты тестирования:

- java-фреймворк для модульного тестирования JUnit;

- инструменты автоматизированного тестирования для функционального

тестирования веб-приложений Selenium;

- инструменты тестирования API, помогающие проверить корректность интерфейса и производительность Postman.

Инструменты развертывания:

- платформа контейнеризации для создания и управления контейнерами для приложений Docker;

- инструмент оркестровки контейнеров, поддерживающий автоматическое развертывание, масштабирование и управление контейнеризированными приложениями Kubernetes;

- инструмент непрерывной интеграции/непрерывного развертывания CI, поддерживающий автоматизированные процессы сборки и развертывания Jenkins.

1.4.3 Методы разработки информационных систем

Методы разработки необходимы для проектирования, создания и внедрения программных продуктов. Они определяют, как организовать рабочий процесс, распределить задачи между командой и обеспечить качественный результат.

Основные методы разработки:

- каскадная модель (Waterfall) представляет классический метод, при котором все этапы разработки проходят последовательно: от анализа требований до тестирования и внедрения. Каждая фаза завершается до начала следующей;

- итеративная и инкрементная разработка используется когда процесс разбивается на небольшие циклы (итерации). В каждой из них выполняются анализ, проектирование, разработка и тестирование. Это позволяет постепенно добавлять новые функции.

- гибкая методология, акцентирующая внимание на взаимодействии с заказчиком, постоянной обратной связи и адаптации к изменениям Agile. Разработка проходит итерациями (спринтами), каждая из которых завершается демонстрацией рабочего продукта;

- частный случай Agile который называется Scrum. Работы делятся на

короткие итерации (спринты) длительностью 1–4 недели. Упор делается на командную работу, прозрачность процессов и гибкость.

- метод управления рабочим процессом, при котором задачи визуализируются на доске Kanban. Уделяется внимание равномерной загрузке команды и непрерывной поставке результата.

- подход, сфокусированный на высоком качестве кода и активном взаимодействии с пользователем называемый экстремальным программированием (XP). Методы включают парное программирование, непрерывное тестирование и частые релизы;

- метод, сочетающий элементы каскадной и итеративной моделей называемый спиральная модель.

Разработка спиральной модели проходит поэтапно, при этом каждый этап завершает виток спирали:

- анализ рисков;
- проектирование;
- разработка;
- тестирование.

Для реализации системы применяется итеративная методология Agile, которая позволяет быстро реагировать на изменения требований заказчика. Это особенно важно в условиях меняющегося рынка. Используются инструменты, упрощающие разработку и тестирование, такие как Spring Boot для серверной части и React для фронтенда.

Для изучения особенностей каждого из методов необходимо составить сравнительную таблицу. Сравнительная таблица методов разработки представлена ниже.

Таблица 1 – Сравнительная таблица методов разработки

Метод	Особенности	Преимущества	Недостатки	Применение
1	2	3	4	5
Каскадная модель	Последовательный процесс без возврата к предыдущим этапам.	Простота управления, четкость этапов.	Невозможность адаптации к изменениям, ошибки выявляются на поздних стадиях.	Малые проекты с четкими требованиями.
Итеративная	Разработка ведется циклами, продукт постепенно улучшается.	Возможность корректировки, быстрый выпуск рабочей версии.	Требуется больше времени и ресурсов, чем при каскадной модели.	Сложные проекты, где важна обратная связь.
Agile	Гибкий подход, основанный на спринтах и активной коммуникации с заказчиком.	Высокая адаптивность, качественная обратная связь, постоянные улучшения.	Требуется высокой вовлеченности команды и заказчика, сложность управления крупными проектами.	Динамичные проекты, стартапы.
Scrum	Подразумевает регулярные встречи (митинги), планирование спринтов и демонстрацию результатов.	Прозрачность работы команды, быстрые результаты.	Не подходит для маленьких команд, сложность внедрения.	Командные проекты с четкими дедлайнами.

Продолжение таблицы 1

1	2	3	4	5
Kanban	Использует визуальную доску для управления процессом.	Простота реализации, подходит для непрерывной работы.	Может быть неэффективен для задач, требующих четких сроков.	Техническая поддержка, задачи без сроков.
XP	Упор на качество кода, частые релизы и тестирование.	Высокое качество ПО, быстрая адаптация к изменениям.	Требует высокой квалификации разработчиков, интенсивность работы может приводить к выгоранию.	Стартапы, где важна стабильность ПО.
Спиральная модель	Комбинация каскадного и итеративного подходов, акцент на анализе рисков.	Учет рисков, гибкость, возможность постепенного развертывания.	Сложность управления, высокие затраты на реализацию.	Сложные долгосрочные проекты.

Для системы продаж в магазинах цветов и птиц на основе Java наиболее подходящим подходом является итеративная и инкрементная разработка или Agile. Это связано с тем, что проект имеет ограниченные временные рамки. Он включает четко определенные функциональные модули (например, управление запасами, учет продаж, работа с клиентами), которые можно разрабатывать постепенно. Требуется демонстрация промежуточных результатов научному руководителю или комиссии.

Постепенная разработка – проект можно разбить на этапы (итерации), каждый из которых фокусируется на разработке и тестировании одного модуля системы, например:

- создание модуля управления запасами;
- реализация учета продаж;
- добавление функциональности работы с клиентами и генерации отчетов.

Возможность корректировок – после каждой итерации можно вносить изменения, если требования уточняются, или добавлять новые функции.

Демонстрация промежуточных результатов – возможность на каждом этапе показывать научному руководителю функциональную часть системы, что ускорит получение обратной связи и позволит своевременно вносить коррективы.

Фокус на приоритетах – можно начать с самых важных модулей, обеспечивая минимально работающий продукт на ранних стадиях.

1.5 Обзор существующих методов решения аналогичных задач

Решения для автоматизации продаж в магазинах цветов и птиц можно разделить на три основные категории:

- ручные методы управления (традиционные подходы);
- программные системы управления (локальные системы);
- облачные и интегрированные решения.

Ручные методы управления – это традиционные подходы, включающие использование бумажных записей или электронных таблиц (например, Excel) для учета товаров, продаж и клиентов.

Преимущества:

- простота внедрения, не требует специализированного оборудования или программного обеспечения;
- низкая стоимость на начальном этапе.

Недостатки:

- высокая вероятность ошибок, ручной ввод данных приводит к нестыковкам;
- ограниченные аналитические возможности: трудоемкий анализ данных;
- невозможность автоматизации повторяющихся процессов;
- низкая скорость обработки информации при росте объема данных.

Примером являются малые магазины, где учет продаж ведется в тетрадях или в файлах Excel. А также программные системы управления (локальные системы) являются программным обеспечением, установленном на компьютерах магазина. Такие системы помогают автоматизировать операции учета запасов,

управления продажами, обработки заказов и формирования отчетов.

Преимущества:

- высокая скорость обработки данных;
- надежность, система работает локально, без зависимости от интернета;
- возможности персонализации под конкретные нужды бизнеса.

Недостатки:

- высокие затраты на разработку, внедрение и поддержку;
- ограниченная доступность, работа возможна только на определенных устройствах;
- необходимость обучения сотрудников.

Примером предлагаемых решений является:

- популярное решение для автоматизации продаж и учета в небольших бизнесах 1С: Управление торговлей. А также локальные кассовые системы, разработанные на заказ;

- облачные и интегрированные решения являются современными системами, работающими через интернет. Они объединяют функции учета запасов, продаж, управления клиентами и анализа данных в едином интерфейсе. Такие решения часто работают по модели SaaS (программное обеспечение как услуга).

2 АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РЕШЕНИЯ ЗАДАЧИ ПРОДАЖ

2.1 Анализ осуществимости

2.1.1 Экономическое обоснование

Целью данной работы является анализ экономической целесообразности создания системы продаж для магазинов цветов и птиц на базе Java для удовлетворения рыночного спроса и повышения эффективности работы предприятия. Целесообразность создания системы и важность повышения конкурентоспособности предприятия объясняются с помощью анализа рынка и проектирования функций системы. После оценки экономической целесообразности результаты показывают, что система имеет высокие экономические и социальные преимущества и обладает потенциалом для коммерческой эксплуатации. Результаты исследования в данной диссертации обеспечивают сильную теоретическую и практическую поддержку для широкого развития и продвижения системы продаж в магазинах цветов и птиц на базе Java, что помогает повысить общий уровень развития и коммерческую ценность рынка цветов и птиц.

2.1.2 Эксплуатационная осуществимость

В данной диссертации анализируется эксплуатационная пригодность системы продаж магазина цветов и птиц на базе Java [9]. Система использует язык программирования Java и реализует функции управления заказами, запасами, пользователями и товарами. Проанализировав код и продемонстрировав процесс работы, выяснилось, что система обладает хорошей эксплуатационной пригодностью. В работе выдвигаются предложения по дальнейшему совершенствованию системы, такие как упрощение процесса работы, улучшение пользовательского опыта и усиление безопасности. Результаты эксплуатационных испытаний показали, что система обладает хорошей эксплуатационной пригодностью, проста в эксплуатации и удобна для пользователя, что повышает эффективность продаж и удовлетворенность клиентов магазина. Данное исследование оказывает серьезную поддержку продвижению и развитию системы продаж в магазинах цветов и

птиц на базе Java, которая повышает эффективность управления и экономическую выгоду от продаж в магазинах.

Для успешной работы информационной системы важно, чтобы интерфейс был интуитивно понятным и удобным для сотрудников магазина. Это включает:

- основные функции (поиск товара, оформление заказа, учет складских запасов) должны быть легко доступны;
- корректная работа на разных устройствах (ПК, планшетах, мобильных телефонах);
- встроенные подсказки и валидация данных, которые помогут избежать неправильного ввода.

Так как система предназначена для сотрудников магазина, потребуется минимальное обучение. Оно может включать:

- инструкции и руководства по использованию системы;
- краткие тренинги для продавцов и администраторов;
- доступная служба поддержки для решения возникающих вопросов.

Информационная система должна органично интегрироваться в существующую работу магазина:

- система управления складом должна поддерживать текущее ведение учета товаров;
- процессы продаж поддерживают взаимодействие с кассовым оборудованием (если необходимо) и учет клиентских заказов;
- отчетность необходима для генерации отчетов для анализа продаж и управления товарными остатками.

Поддержка и сопровождение включают в себя:

- доступность технической поддержки для устранения возможных сбоев;
- обновления и доработки по мере необходимости;
- резервное копирование данных для предотвращения потери информации.

Эксплуатационная осуществимость информационной системы «Магазин продажи цветов и птиц» оценивается как высокая. Система будет интуитивно понятной, легко интегрируемой в существующие бизнес-процессы и требующей

минимального обучения персонала. Поддержка и сопровождение обеспечат бесперебойную работу, а удобный интерфейс повысит эффективность работы сотрудников.

2.1.3 Техническая осуществимость

Данная система разработана на основе Java, Spring Boot, Spring, Thymeleaf в качестве динамического представления данных, Maven для управления зависимостями и репозиториумом пакетов, IntelliJ IDEA 2019.1.3 как среда разработки.

Подключение к базе данных MySQL осуществляется через JDBC, с использованием стандартного языка структурированных запросов SQL для хранения и управления данными.

Для развертывания системы требуется:

- ОС Linux (Ubuntu, CentOS) или Windows Server, процессор от 2,5 ГГц, оперативная память 8 ГБ, SSD-диск от 256 ГБ;
- ПК или ноутбуки с Windows/Linux/macOS, процессором от 2 ГГц, 4 ГБ ОЗУ, браузерами Chrome, Firefox, Edge;
- поддержка HTTPS, SSL/TLS для защиты передаваемых данных.

Технологический стек включает в себя:

- Java, Spring Boot, Spring MVC, Hibernate (для работы с базой данных);
- Thymeleaf, HTML, CSS, JavaScript (для динамического отображения данных);
- MySQL, поддерживающая ACID-транзакции и репликацию;
- Git (GitHub/GitLab/Bitbucket).

Основные методы исследования, использованные при разработке и внедрении сайта:

- изучение научных и технических источников, связанных с информационными системами и онлайн-продажами.
- анализ работы существующих платформ для онлайн-продаж цветов и птиц, выявление ключевых функциональных требований.
- моделирование работы системы, тестирование ее производительности, надежности и безопасности, внесение корректировок для оптимизации.

Система может быть интегрирована со следующими сервисами:

- платежными сервисами (например, ЮKassa, PayPal, Stripe) для онлайн-оплаты;
- системами складского учета для автоматического обновления остатков товаров;
- сервисами аналитики (Google Analytics, Яндекс.Метрика) для мониторинга активности пользователей.

Обеспечение безопасности включает в себя:

- аутентификация и авторизация пользователей (Spring Security, JWT, OAuth 2.0);
- шифрование конфиденциальных данных (например, пароли хранятся в хэшированном виде с использованием BCrypt);
- регулярное резервное копирование базы данных для предотвращения потери данных.

2.1.4 Схема системы

Разработка информационной системы для магазина цветов и птиц основана на тщательном анализе бизнес-процессов, связанных с продажами и маркетингом. В результате анализа была создана системная блок-схема, отражающая ключевые модули системы и их взаимодействие.

На рисунке 2 показана системная схема системы продаж магазина цветов и птиц на базе Java.

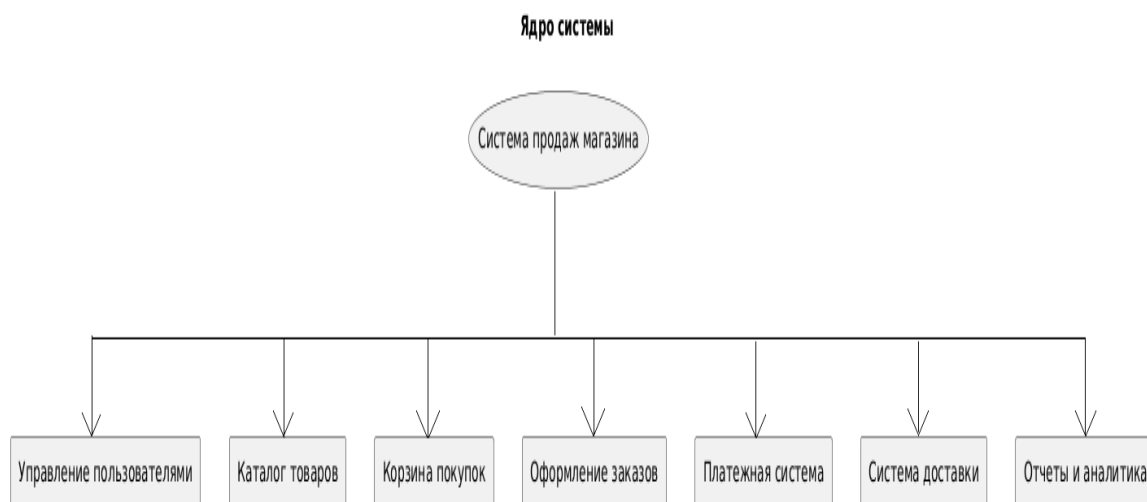


Рисунок 2 – Схема системы продаж магазина цветов и птиц на базе Java

Созданная схема представляет собой модульную архитектуру, где каждый компонент отвечает за определённую функциональность.

Основные модули системы:

- управление пользователями включает таблицу пользователей и их данные, обеспечивает регистрацию, аутентификацию и авторизацию;
- определение ролей реализует разграничение прав пользователей (администратор, менеджер, покупатель);
- управление товарами обеспечивает добавление, изменение, удаление и поиск информации о товарах, включает базу данных с товарами (цветами и птицами);
- управление корзиной позволяет добавлять, изменять и удалять товары из корзины, включает механизм расчета итоговой стоимости заказа;
- управление заказами позволяет оформлять, редактировать и удалять заказы, хранит историю покупок клиентов;
- управление отзывами обеспечивает функционал добавления, редактирования и удаления отзывов о товарах;
- управление обратной связью позволяет клиентам оставлять вопросы и

запросы, содержит базу данных с запросами пользователей.

Принципы работы системы:

- каждый компонент выполняет четко определенные функции;
- возможность масштабирования и добавления новых функций;
- аутентификация и авторизация пользователей, защита данных;
- оптимизированные SQL-запросы для работы с базой данных MySQL.

Разработанная схема демонстрирует логическую организацию системы, обеспечивая надежную и удобную платформу для онлайн-продаж. Она легко адаптируется под изменения в бизнес-процессах и может быть дополнена новыми функциями.

2.2 Анализ потребностей

2.2.1 Функциональные требования

Java основе цветов и птиц магазин продаж система в основном используется для цветов и птиц магазин продаж бизнес-информации операции, цветов и птиц магазин продаж системы могут быть зарегистрированы, в соответствии с различными типами пользователей для выполнения различных операций, например, администратор может быть основан на фактической ситуации настройки или изменения типа пользователя каждого пользователя, пользователь вошел в соответствии с администратором типа пользователя может быть основана на тип пользователя настроен можно увидеть в различных вариантах, администратор может работать все варианты. Администратор может управлять всеми опциями и управлять информацией о пользователе, информацией о продукте, информацией о заказе, информацией о типе, информацией о корзине, информацией о клиенте, информацией о продавце, информацией об оценке и информацией об отзывах.

Система будет отслеживать текущие запасы цветов, птиц, аксессуаров и других товаров. В случае нехватки товара или при поступлении новых поступлений система будет автоматически обновлять информацию о наличии товаров в магазине и предоставлять уведомления для персонала о необходимости пополнения ассортимента.

Информационная система будет включать удобный и интуитивно понятный каталог, где пользователи смогут ознакомиться с полным ассортиментом товаров, включая подробные описания, фотографии, цены и количество в наличии. Товары будут классифицированы по категориям (цветы, птицы, аксессуары и т. д.).

Система обеспечит управление заказами клиентов, включая возможность оформления и оплаты заказов через различные каналы. После оформления заказа пользователю будет предоставлена информация о статусе его выполнения (принят, в обработке, готов к отправке или доставке).

Информационная система будет интегрирована с различными платёжными системами для удобства оплаты, включая онлайн-банкинг, карты, электронные кошельки и наличные. Процесс оплаты будет защищён с помощью современных технологий безопасности.

Система будет поддерживать функционал регистрации и учёта клиентов. Каждый покупатель будет иметь свой профиль, в котором хранится история покупок, предпочтения, возможные скидки и накопленные бонусы. Это позволит персонализировать предложения и улучшить взаимодействие с клиентами.

Для привлечения клиентов система позволит запускать различные акции и скидки, например, на определённые группы товаров, для постоянных клиентов или в рамках сезонных распродаж. Скидки могут быть настроены как для определённых товаров, так и для общей суммы покупки.

Система будет поддерживать автоматизированное управление поставками товаров от поставщиков. Для каждого товара будет доступна информация о его стоимости, сроках поставки и поставщиках. Это позволит оптимизировать процесс закупок и своевременно пополнять запасы.

Система будет генерировать отчёты и статистику по продажам, доходам, популярности товаров, эффективности акций и другим важным показателям. Эти данные будут использоваться для принятия решений по улучшению бизнес-процессов и прогнозированию будущих продаж.

В системе будет предусмотрена возможность управления сотрудниками

магазина: учёт рабочего времени, настройка ролей и прав доступа, мониторинг выполнения задач. Это повысит эффективность работы сотрудников и улучшит организацию процессов в магазине.

Для расширения функционала система будет интегрироваться с внешними сервисами, такими как службы доставки, бухгалтерские и складские системы, а также маркетинговые платформы. Это позволит оптимизировать работу магазина и улучшить взаимодействие с партнёрами.

Для более детального описания архитектуры в отчете используются три ключевых типа диаграмм, которые служат инструментами для визуализации структуры системы и демонстрации различных аспектов её работы. Эти диаграммы включают диаграмму прецедентов, диаграмму последовательности и диаграмму состояний.

Диаграмма состояний предназначена для демонстрации динамического поведения системы, отображая изменения состояний объектов и взаимодействие с ними через события и действия. В рамках проекта FinancialSystem диаграмма состояний описывает поведение ключевых компонентов системы в зависимости от внутренних и внешних факторов, таких как изменения данных, действия пользователя или административные операции.

Диаграмма прецедентов представлена на рисунке 3.

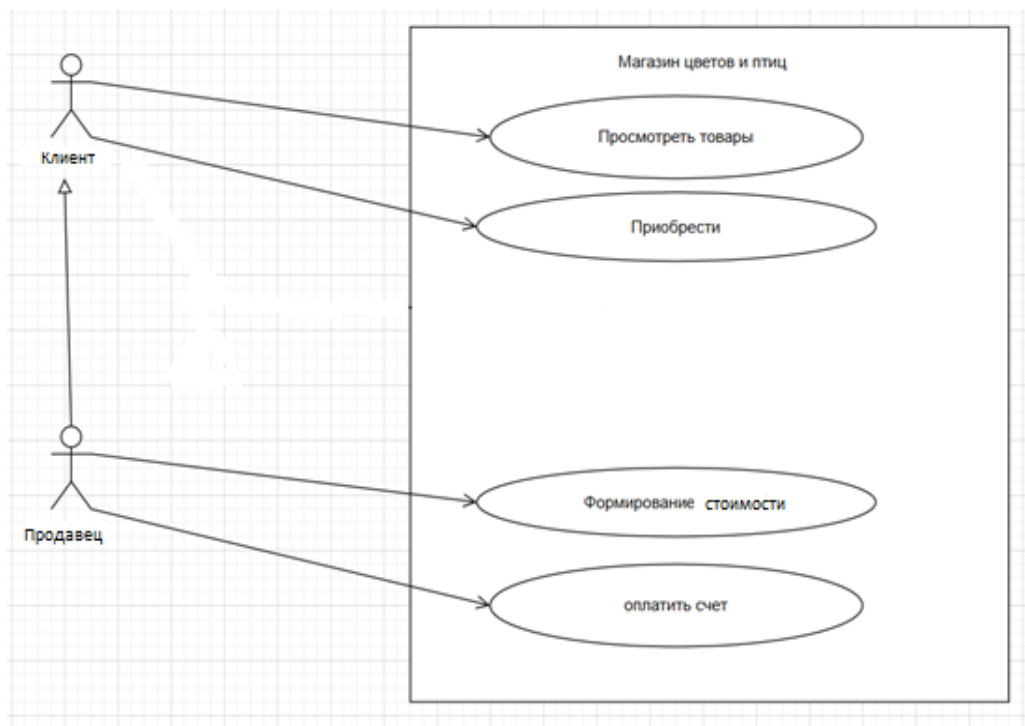


Рисунок 3 – Диаграмма прецедентов

Проектирование информационной системы «Магазин продажи цветов и птиц» с использованием UML (Unified Modeling Language) предполагает создание различных диаграмм для визуализации структуры и процессов системы. Включает в себя такие элементы, как диаграммы классов, которые отображают основные сущности, такие как «Цветы», «Птицы», «Клиенты», «Заказы», «Поставки» и их атрибуты, а также связи между ними. Диаграммы последовательностей и активности помогают описать динамику взаимодействия пользователей с системой, процесс оформления заказа, обработки платежей и выполнения поставок. Важно также учитывать систему учета товарных запасов, прогнозирование потребностей в поставках и отчеты о продажах. UML помогает структурировать и анализировать требования, обеспечивая четкую картину как функционирует система в целом [39].

2.2.2 Требования к характеристикам

Целью требований к целостности системы является обеспечение функционирования всех основных модулей системы, таких как:

- вход в систему;

- информация о пользователе;
- информация о продукте;
- информация о заказе;
- информация о типе продукта;
- информация о корзине;
- информация о клиенте;
- информация о продавце;
- информация об оценке;
- управление отзывами.

Эти функции обеспечивают работу системы продаж магазина «Цветок и птица» на базе Java.

Информационная система для магазина продажи цветов и птиц предназначена для автоматизации основных бизнес-процессов: управления товарными запасами, обработки заказов клиентов, ведения финансовых операций и отчетности, а также для улучшения взаимодействия с клиентами. Важно, чтобы система была гибкой и масштабируемой, поддерживая расширение ассортимента товаров и возможность подключения дополнительных каналов продаж (например, интернет-магазин). Архитектура системы должна быть модульной, обеспечивая независимость различных компонентов друг от друга.

Система будет построена на клиент-серверной архитектуре. Серверная часть будет выполнять все ключевые операции по обработке данных и управлению бизнес-логикой, в то время как клиентская часть (например, веб-интерфейс или мобильное приложение) будет предоставлять доступ пользователям. В качестве базы данных рекомендуется использовать реляционную СУБД (например, MySQL или PostgreSQL), которая позволит эффективно управлять большими объемами данных о товарах, заказах и клиентах. Для обеспечения безопасности и защиты данных будет реализована система аутентификации и авторизации пользователей.

Система должна поддерживать интеграцию с внешними сервисами, такими как платёжные шлюзы, системы доставки и онлайн-оплаты. Также

необходимо предусмотреть возможность интеграции с другими программными решениями магазина, например, системой управления складом или бухгалтерией. Для этого будут использованы открытые API и стандарты обмена данными. Архитектура должна быть гибкой, чтобы в будущем можно было легко расширить функционал (например, добавить модуль для онлайн-торговли или интеграцию с CRM-системой для управления взаимоотношениями с клиентами).

Проектирование информационной системы «Магазин продажи цветов и птиц» включает создание последовательных диаграмм (sequence diagrams), которые отображают взаимодействие между различными компонентами системы [11]. Эти диаграммы показывают порядок обмена сообщениями между объектами, что помогает визуализировать процессы, такие как оформление заказа, управление товарным запасом и обработка платежей, обеспечивая эффективную работу системы.

Диаграмма последовательности представлена на рисунке 4.

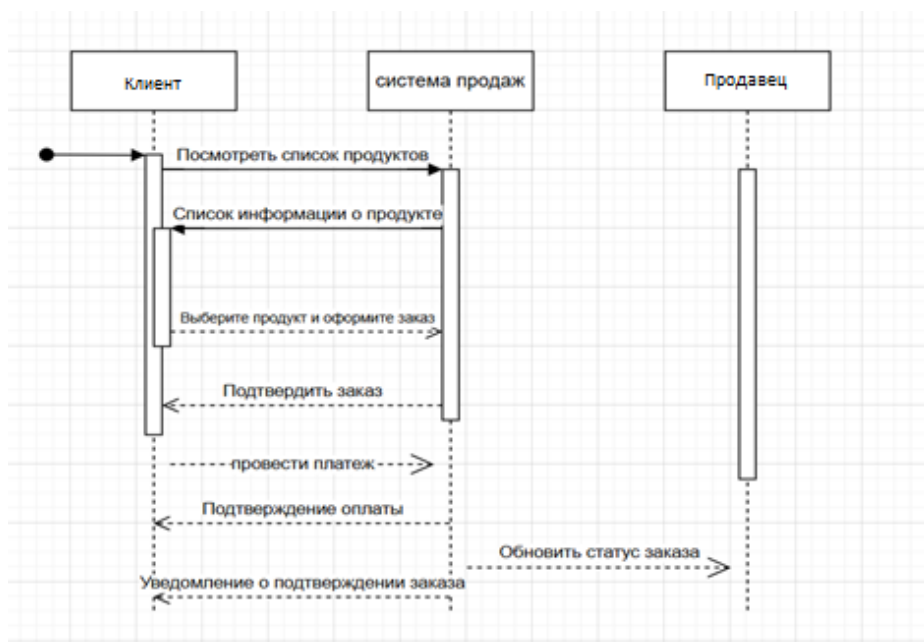


Рисунок 4 – Диаграмма последовательности

Проектирование информационной системы «Магазин продажи цветов и птиц» с использованием диаграмм последовательности позволяет детально описать взаимодействие между пользователями и системой на различных этапах процесса, таких как покупка товара, оформление заказа и обработка платежей.

Эти диаграммы помогают визуализировать последовательность операций, что способствует ясному пониманию бизнес-логики и улучшению функциональности системы. В результате использование диаграмм последовательности помогает создать более эффективную и удобную в эксплуатации систему, обеспечивающую плавное и логичное взаимодействие всех компонентов.

Целью требований к безопасности системы является обеспечение безопасности данных, хранящихся в системе.

Для этого используется СУБД MySQL, которая обеспечивает точность и безопасность данных [28].

Методы:

- установка соединения с базой данных через ввод правильного имени пользователя и пароля;
- обеспечение защиты сети для предотвращения несанкционированного доступа.

Целью требований к работоспособности системы является обеспечение удобства использования системы для всех пользователей, независимо от их уровня технической подготовки.

Методы:

- портативность системы, чтобы она могла легко использоваться на различных устройствах;
- простота эксплуатации, благодаря удобному интерфейсу и соответствующим подсказкам в процессе работы.

Это особенно важно, поскольку система предназначена для магазинов цветов и птиц, и должна быть легко понятной и доступной для использования широким кругом пользователей.

2.3 Обзор возможностей профильного программного обеспечения

В данном подразделе необходимо рассмотреть программные средства, которые могут быть использованы при проектировании и разработке информационной системы «Магазин продажи цветов и птиц».

2.3.1 Платформы и фреймворки

При создании информационной системы «Магазин продажи цветов и птиц» важно выбрать оптимальный стек технологий. Основным языком программирования для разработки – Java, а для ускорения и упрощения процесса используется Spring Boot.

Java – один из самых популярных языков программирования, который широко используется для разработки корпоративных систем, включая веб-приложения.

Преимущества Java для проекта:

- возможность развертывания приложения на Windows, Linux, macOS;
- встроенная система управления памятью, защита от утечек данных;
- разработка как небольших, так и крупных систем;
- множество библиотек и инструментов для веб-разработки.

Spring Boot является фреймворком для создания веб-приложений на Java, который позволяет быстро разрабатывать и развертывать микросервисные и монолитные приложения.

Преимущества Spring Boot:

- минимальные настройки для запуска приложения;
- сервер, не требующий развертывания;
- удобная работа с базами данных Spring Data JPA;
- встроенные механизмы аутентификации и авторизации Spring Security;
- архитектура легко интегрирующаяся с другими сервисами.

Для динамического отображения данных на веб-страницах используется Thymeleaf.

Преимущества:

- полная интеграция с Spring Boot;
- позволяет писать HTML-код с динамическими элементами;
- работает как на стороне сервера, так и в статическом режиме.

Хотя основная логика выполняется на сервере, фронтенд играет важную роль в пользовательском интерфейсе.

Используемые технологии:

- разметка страниц HTML5;
- стилизация элементов, адаптивный дизайн CSS3;
- динамическое обновление контента JavaScript;
- обновление данных без перезагрузки страницы AJAX.

Если проекту потребуется более интерактивный интерфейс, можно рассмотреть React или Vue.js для фронтенда.

Maven – инструмент, который автоматизирует процесс сборки проекта и управления зависимостями.

Функции Maven:

- автоматическая загрузка библиотек;
- управление версиями зависимостей;
- унифицированная структура проекта.

Таким образом, использование Java, Spring Boot, Thymeleaf, MySQL обеспечивает:

- быструю разработку и масштабируемость.
- безопасность данных (Spring Security);
- гибкость в развертывании (Docker, облачные сервисы);
- интуитивно понятный интерфейс для пользователей.

Этот стек позволяет построить надежную и современную систему электронной коммерции.

2.3.2 Среда и инструменты разработки

Для успешной разработки информационной системы «Магазин продажи цветов и птиц» используется ряд профессиональных инструментов, облегчающих написание кода, тестирование, управление версиями и развертывание.

IntelliJ IDEA – одна из самых мощных сред разработки для Java, поддерживающая Spring Boot и другие современные технологии.

Преимущества:

- поддержка Spring Boot (автоматические подсказки, встроенный Spring Initializr);

- умный автодополнительный код (AI Code Completion);
- встроенный отладчик и профилировщик;
- интеграция с Git, Docker, базами данных;
- поддержка Maven и Gradle для управления зависимостями.

Внешний вид IntelliJ IDEA представлен на рисунке 5.

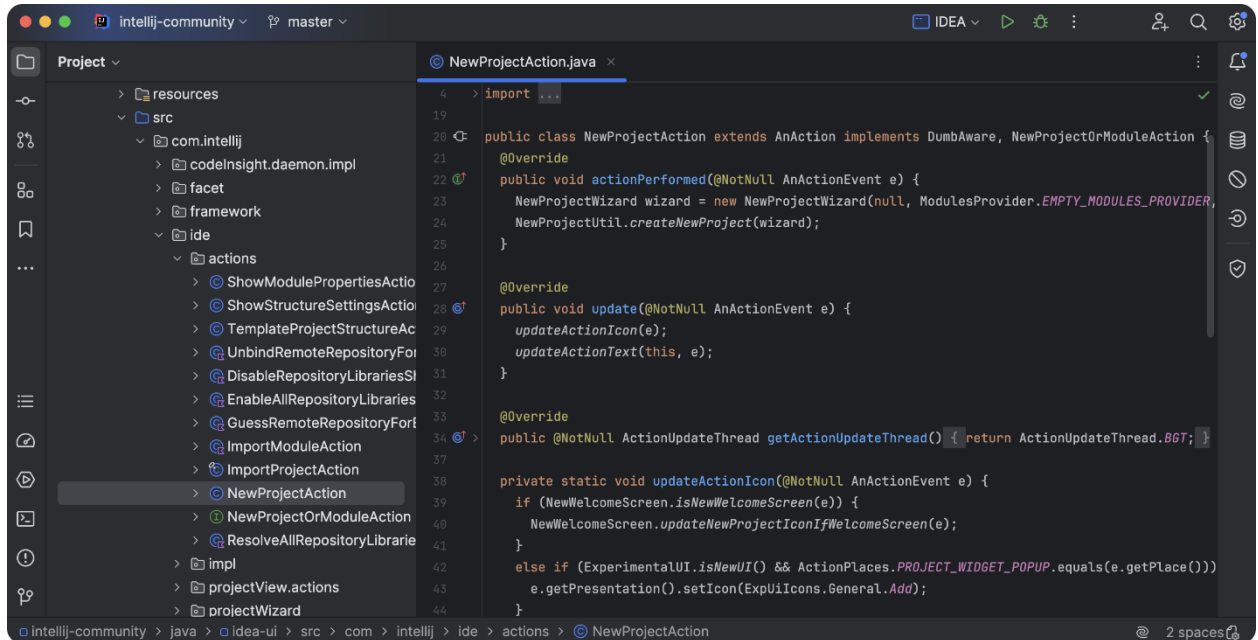


Рисунок 5 – Интерфейс IntelliJ IDEA

В Таблице 2 представлена сравнительная информация о существующих инструментах разработки.

Таблица 2 – Сравнительная таблица инструментов разработки

Критерий	IntelliJ IDEA	Eclipse	NetBeans
1	2	3	4
Производительность	Высокая, требует больше ресурсов, но работает быстрее благодаря индексации кода	Средняя, иногда подтормаживает при работе с большими проектами	Средняя, но менее оптимизирована

Продолжение таблицы 2

1	2	3	4
Поддержка Java	Отличная, встроенная поддержка всех версий Java, включая Spring Boot	Хорошая, но требует установки плагинов для некоторых функций	Хорошая, но медленнее обновляется
Поддержка Spring Boot	Полная, встроенный Spring Initializr, удобная работа с аннотациями	Требует установки дополнительных плагинов	Ограниченная, неудобная работа с конфигурацией
Автодополнение кода	Интеллектуальное, использует AI для предсказания кода	Простое автодополнение, менее точное	Среднее, не всегда работает корректно
Отладка (Debugging)	Мощный встроенный отладчик, удобная работа с breakpoint'ами	Достаточно хороший, но менее удобный интерфейс	Основные функции есть, но интерфейс перегружен
Поддержка рефакторинга	Очень развитые инструменты, безопасное изменение кода	Есть, но работает медленнее и не всегда корректно	Основные функции есть, но ограниченные
Поддержка плагинов	Большой выбор плагинов, включая поддержку других языков	Открытая платформа, огромное количество плагинов	Количество плагинов ограничено
Интеграция с Git	Встроенная поддержка Git, удобный интерфейс	Есть, но менее удобная навигация	Базовая поддержка Git, ограниченный функционал

Продолжение таблицы 2

1	2	3	4
Интеграция с базами данных	Встроенный Database Tool, удобный доступ к SQL	Требуется установка плагинов	Поддержка есть, но менее удобная
Простота использования	Удобный и интуитивно понятный интерфейс	Требуется настройка перед началом работы	Интерфейс устаревший, но простой
Лицензия	Бесплатная Community-версия, платная Ultimate-версия	Полностью бесплатный	Полностью бесплатный

Git – система управления версиями, позволяющая отслеживать изменения в коде и работать в команде.

Основные команды Git:

- git init – инициализация репозитория;
- git add . – добавление файлов в индекс;
- git commit -m «Описание» – фиксация изменений;
- git push origin main – отправка изменений в удаленный репозиторий.

Платформы для хранения кода:

- GitHub – самый популярный сервис, поддерживает pull-requests, CI/CD;
- GitLab – удобен для CI/CD, если проект планируется разворачивать в облаке;
- Bitbucket – подходит для закрытых корпоративных проектов.

Далее приведен пример выбранной системы, предназначенной для управления зависимостями.

Maven – инструмент, который управляет библиотеками, необходимыми для работы проекта.

Преимущества:

- автоматически загружает и обновляет библиотеки;

- управляет зависимостями через файл pom.xml;
- поддержка множества плагинов (тестирование, сборка, развертывание).

Альтернативой является Gradle, он более гибкий инструмент, но менее распространён в Spring Boot-проектах.

В качестве инструмента для работы с базами данных используется MySQL Workbench. Он представляет графический инструмент для управления базой данных MySQL.

Функции:

- визуальное проектирование схемы базы данных;
- запуск SQL-запросов и их отладка;
- импорт/экспорт данных.

Для тестирования API используется Postman.

Основные функции:

- позволяет отправлять HTTP-запросы (GET, POST, PUT, DELETE);
- имеет удобный интерфейс для работы с JSON и XML;
- имеет возможность автоматизации тестов API.

Docker – инструмент отвечающий за контейнеризацию приложения. Он позволяет упаковать приложение в контейнер для быстрого развертывания на сервере.

Преимущества:

- работает одинаково на любой системе (Windows, Linux, macOS);
- изолированное окружение (не нужно устанавливать зависимости на сервер);
- поддержка Kubernetes для масштабирования.

В качестве сервера для развертывания используется Apache Tomcat. Это Легковесный веб-сервер, поддерживающий Java-приложения.

Преимущества:

- простота настройки и управления.
- интеграция со Spring Boot.

2.3.3 Архитектурный стиль

При проектировании системы «Магазин продажи цветов и птиц» важно выбрать правильный архитектурный стиль, который обеспечит гибкость, масштабируемость, удобство разработки и поддержки.

Основные архитектурные стили

монолитная архитектура – архитектурный стиль, в котором вся функциональность системы объединена в одно целое приложение. Вся логика, интерфейс и работа с базой данных находятся в одном кодовом проекте. Такой подход прост в разработке, но усложняет масштабирование и поддержку при росте системы.

клиент-серверная архитектура – модель, в которой приложение разделено на интерфейс пользователя, обработку данных и логику. Клиент отправляет запросы серверу, который обрабатывает их и отправляет ответ. Этот подход позволяет разделить нагрузку между устройствами и централизовать управление данными.

многослойная (N-tier) архитектура – расширенная версия клиент-серверной модели, в которой система делится на представление (UI), бизнес-логику и базу данных. Разделение позволяет улучшить масштабируемость, облегчить поддержку и тестирование. Данная архитектура часто используется в корпоративных системах.

микросервисная архитектура – стиль проектирования, при котором приложение разбивается на независимые сервисы, каждый из которых выполняет отдельную функцию и взаимодействует с другими через API. Такой подход обеспечивает гибкость, независимое развертывание и масштабируемость, но требует сложного управления и оркестрации.

Для системы «Магазин продажи цветов и птиц» оптимальным вариантом является многослойная архитектура (N-tier) с применением модели MVC (Model-View-Controller).

Описание многослойной архитектуры (N-tier architecture)

Система разделена на три ключевых уровня:

клиентский уровень (Frontend, UI Layer) – уровень, отвечающий за

взаимодействие с пользователем. Он включает в себя интерфейс, через который пользователь вводит данные и получает результаты [13]. Этот уровень разрабатывается с использованием технологий HTML, CSS, JavaScript и различных фреймворков (например, React, Angular, Vue.js). Клиентский уровень отправляет запросы серверу и отображает полученные данные;

серверный уровень (Backend, Application Layer) – уровень, отвечающий за обработку бизнес-логики приложения. Он получает запросы от клиента, обрабатывает их, взаимодействует с базой данных и отправляет обратно результаты. Серверный уровень разрабатывается с использованием языков программирования, таких как Java (Spring Boot), Python (Django, Flask), Node.js и других;

уровень базы данных (Database Layer) – уровень, предназначенный для хранения, обработки и управления данными. Он принимает запросы от серверного уровня и выполняет операции чтения, записи, обновления и удаления данных. В качестве СУБД могут использоваться реляционные системы (MySQL, PostgreSQL) или NoSQL-хранилища (MongoDB, Redis).

Описание модели MVC (Model-View-Controller).

Система делится на 3 функциональных компонента:

- model (Модель) компонент, отвечающий за управление данными и бизнес-логикой приложения. Модель обрабатывает запросы, взаимодействует с базой данных и предоставляет данные в структурированном виде. В веб-приложениях Model реализуется с помощью классов и ORM (например, Hibernate, Django ORM);

- view (Представление) компонент, отвечающий за отображение информации пользователю. Представление формирует интерфейс на основе данных, полученных от модели, и может включать HTML, CSS, JavaScript и шаблонизаторы (например, Thymeleaf, JSP, Handlebars);

- controller (Контроллер) компонент, который управляет взаимодействием между моделью и представлением. Контроллер принимает входящие запросы от пользователя, вызывает соответствующую логику модели, обрабатывает данные и передает их в представление для отображения. В веб-разработке контроллеры

часто реализуются с использованием Spring MVC, Django Views или Express.js.

Пример использования MVC в системе представлен на рисунке 6.

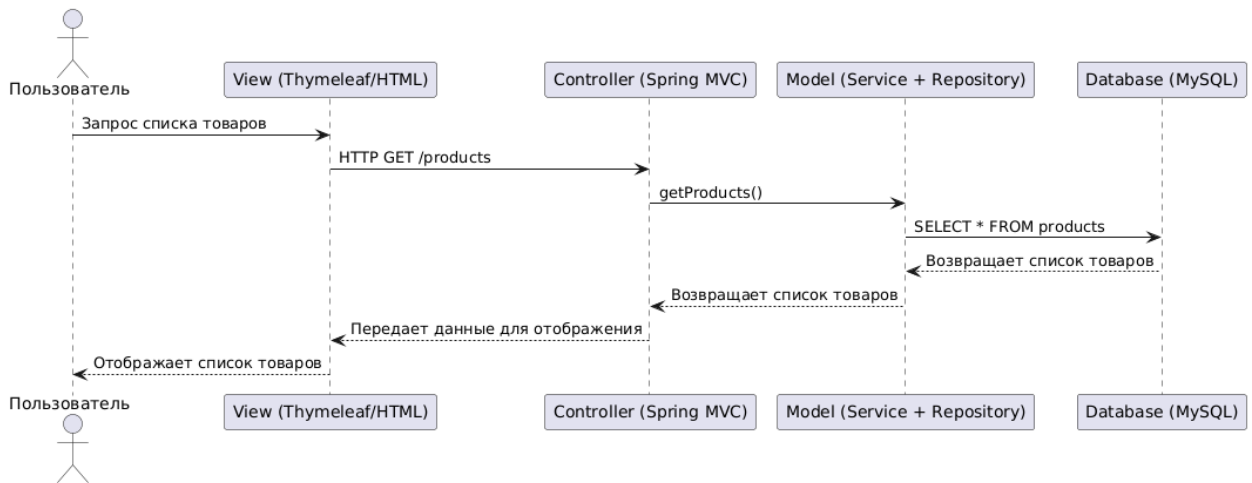


Рисунок 6 – Пример использования MVC

Описание процесса:

- пользователь запрашивает список товаров;
- View (интерфейс) передает запрос в Controller через HTTP-запрос;
- Controller вызывает метод Model для получения данных;
- Model делает SQL-запрос в Database (MySQL);
- Database возвращает данные в Model;
- Model передает список товаров в Controller;
- Controller передает данные в View для отображения;
- View отображает данные пользователю.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРЕДЛАГАЕМОГО АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ ПРОДАЖ

3.1 Основные этапы практической разработки программного продукта

3.1.1 Проектирование архитектуры системы

Проектирование информационной системы «Магазин продажи цветов и птиц» с использованием диаграмм состояний (state diagrams) помогает моделировать изменения состояния объектов системы в ответ на различные события и действия пользователей. Диаграммы состояний описывают жизненный цикл ключевых элементов, таких как заказы, товары или пользователи, и показывают, как эти элементы переходят из одного состояния в другое. Это позволяет наглядно представить процесс обработки заказов, управление товарными запасами и другие важные аспекты, что способствует более точному и эффективному проектированию системы.

Проектирование информационной системы «Магазин продажи цветов и птиц» с использованием диаграмм состояний позволяет подробно проанализировать и отобразить изменения состояний объектов системы в ответ на различные события. Эти диаграммы помогают моделировать поведение ключевых сущностей, таких как заказы, товары и клиенты, на протяжении всего их жизненного цикла. Например, заказ может изменять свои состояния от «создан» до «обработан» и «доставлен», что позволяет точнее отслеживать и управлять процессами внутри магазина.

Использование диаграмм состояний способствует лучшему пониманию логики работы системы, что делает её проектирование более эффективным и прозрачным. Это также помогает избежать ошибок при реализации и тестировании, улучшая взаимодействие между компонентами системы. В целом, диаграммы состояний играют важную роль в создании гибкой и надежной информационной системы, которая точно отражает бизнес-процессы магазина.

На рисунке 7 представлена диаграмма состояния.

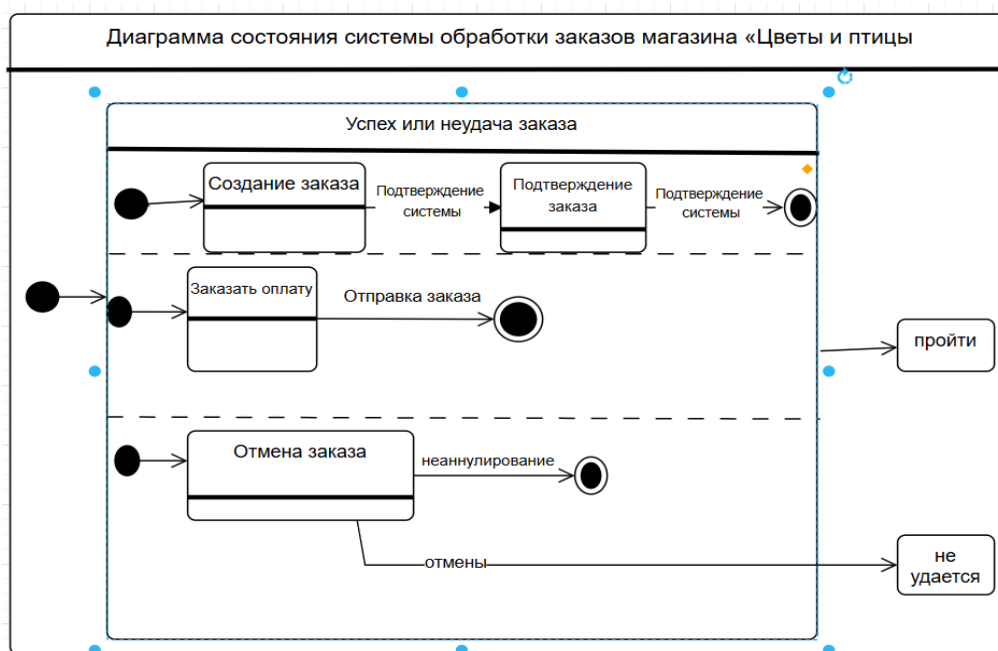


Рисунок 7 – Диаграмма состояния

Проектирование информационной системы «Магазин продажи цветов и птиц» с использованием диаграмм последовательности позволяет детально описать взаимодействие между пользователями и системой на различных этапах процесса, таких как покупка товара, оформление заказа и обработка платежей. Эти диаграммы помогают визуализировать последовательность операций, что способствует ясному пониманию бизнес-логики и улучшению функциональности системы. В результате использование диаграмм последовательности помогает создать более эффективную и удобную в эксплуатации систему, обеспечивающую плавное и логичное взаимодействие всех компонентов.

Реализация программного продукта информационной системы «Магазин продажи цветов и птиц» начинается с разработки архитектуры системы. На этом этапе важно выбрать подходящие технологии и инструменты для создания удобной и функциональной платформы. Для реализации могут быть использованы такие языки программирования, как Java, Python или C#, а также базы данных, например, MySQL или PostgreSQL, для хранения информации о товарах, заказах и клиентах. Архитектура системы должна предусматривать возможности для масштабирования и обработки большого объема данных, а также обеспечивать

высокую степень безопасности и защиты личных данных пользователей.

На следующем этапе осуществляется разработка интерфейсов пользователя (UI/UX). Дизайн должен быть интуитивно понятным и удобным для пользователей, с учетом всех возможных сценариев покупки товаров. Важно создать интерфейс, который будет поддерживать просмотр и заказ цветов и птиц, а также управление корзиной покупок, оформлением заказов и оплатой. Платформа должна быть адаптивной и работать как на десктопных компьютерах, так и на мобильных устройствах.

Затем происходит разработка серверной части системы. Это включает в себя создание логики для обработки запросов, управления товарами, обработки заказов и платежей. Сервер должен обеспечивать быстрый отклик и бесперебойную работу системы даже при высоких нагрузках. Важно также интегрировать систему с платёжными шлюзами для обработки финансовых транзакций, а также предусмотреть функционал для генерации отчетности и анализа данных, таких как статистика продаж и учет товарных остатков.

Завершающим этапом является тестирование и внедрение системы. На этом этапе проводятся функциональные, нагрузочные и безопасности тесты для проверки корректности работы системы и выявления возможных уязвимостей. После успешного тестирования продукт внедряется в эксплуатацию, и начинается этап обучения персонала и пользователей. Важно также предусмотреть регулярные обновления системы и поддержку, чтобы обеспечить стабильную работу и вовремя устранять любые возникающие проблемы.

При создании заказа сервер получает запрос от клиента, содержащий информацию о выбранных товарах, их количестве и возможных дополнительных параметрах (например, сорт цветов или порода птиц). Сервер проверяет наличие этих товаров на складе с помощью соответствующих запросов к базе данных, где хранится актуальная информация о наличии товаров и их количестве.

После проверки доступности товаров сервер производит расчет общей стоимости заказа, учитывая цены товаров, возможные скидки, налоги и расходы на доставку. Этот расчет может зависеть от ряда факторов, таких как выбранная дата

доставки или конкретные условия акций для определенных категорий клиентов. После этого заказ готов к подтверждению, и данные о нем сохраняются в базе данных для дальнейшей обработки, включая создание счетов и генерацию документов для отправки клиенту.

После завершения оформления заказа сервер взаимодействует с базой данных для сохранения информации о новом заказе, статусе его обработки и оплате. Все эти данные будут использоваться для дальнейшего взаимодействия с клиентом (например, отправка уведомлений о статусе заказа), а также для анализа статистики и учета продаж в магазине. На рисунке 8 представлен код для создания заказа.

```
@app.route('/orders', methods=['POST'])
def create_order():
    data = request.get_json()
    customer_name = data.get('customer_name')
    customer_email = data.get('customer_email')
    product_id = data.get('product_id')

    product = Product.query.get_or_404(product_id)
    order = Order(customer_name=customer_name, customer_email=customer_email, product_id=product.id)

    db.session.add(order)
    db.session.commit()

    return jsonify({'order_id': order.id, 'status': order.status}), 201
```

Рисунок 8 – Создать заказ

Проектирование информационной системы для «Магазина продажи цветов и птиц» начинается с создания структуры данных и инициализации всех необходимых компонентов, которые будут поддерживать функционирование системы. На первом этапе важно определить основные сущности и создать их базовые компоненты, такие как базы данных и подключаемые библиотеки.

На рисунке 9 представлен фрагмент кода, взятый из библиотеки импорта.

```
from flask import Flask, jsonify, request
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///store.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)
```

Рисунок 9 – Библиотека импорта

Проектирование информационной системы для магазина, специализирующегося на продаже цветов и птиц, требует разработки комплексной модели данных, которая будет поддерживать эффективное управление процессами покупки, продажи, учета товаров, заказов и взаимодействия с клиентами [7]. На рисунке 10 представлен фрагмент кода, взятый из модели данных.

```
class Product(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    price = db.Column(db.Float, nullable=False)
    category = db.Column(db.String(50), nullable=False)

class Order(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    customer_name = db.Column(db.String(100), nullable=False)
    customer_email = db.Column(db.String(100), nullable=False)
    product_id = db.Column(db.Integer, db.ForeignKey('product.id'), nullable=False)
    product = db.relationship('Product', backref='orders', lazy=True)
    status = db.Column(db.String(50), default='Pending')
```

Рисунок 10 – Модель данных

Структура проекта:

- settings.py параметры проекта, настройки базы данных, аутентификации и других глобальных настроек;
- urls.py маршрутизация URL, определение доступных маршрутов и связи их с соответствующими представлениями;

- `wsgi.py` и `asgi.py` развертывание в продуктивной и асинхронной средах;
- `models.py` модели данных, которые будут представлять товары, заказы, клиентов, поставщиков и другие сущности, связанные с бизнесом;
- `views.py` обработка запросов, например, отображение списка доступных товаров (цветов и птиц), оформление заказов, обработка покупок;
- `serializers.py` сериализация данных в JSON, особенно полезно для работы с API и мобильными приложениями;
- `urls.py` определение маршрутов, которые будут использоваться для API и веб-страниц;
- `admin.py` настройка административной панели, с помощью которой можно будет управлять данными о товарах, клиентах и заказах.

Такая организация системы позволяет эффективно вносить изменения, масштабировать проект и добавлять новые функции для улучшения работы магазина продажи цветов и птиц.

Легкость в поддержке – четкая структура директорий и файлов помогает разработчикам быстро ориентироваться в проекте, что делает его удобным для работы в команде.

Использование шаблонов и данных – возможность легко обновлять и наполнять каталог товарами, а также генерировать отчеты для анализа продаж.

3.1.2 Разработка структуры системы

На рисунке 11 показаны функциональные возможности, реализованные системой продаж в магазинах цветов и птиц.

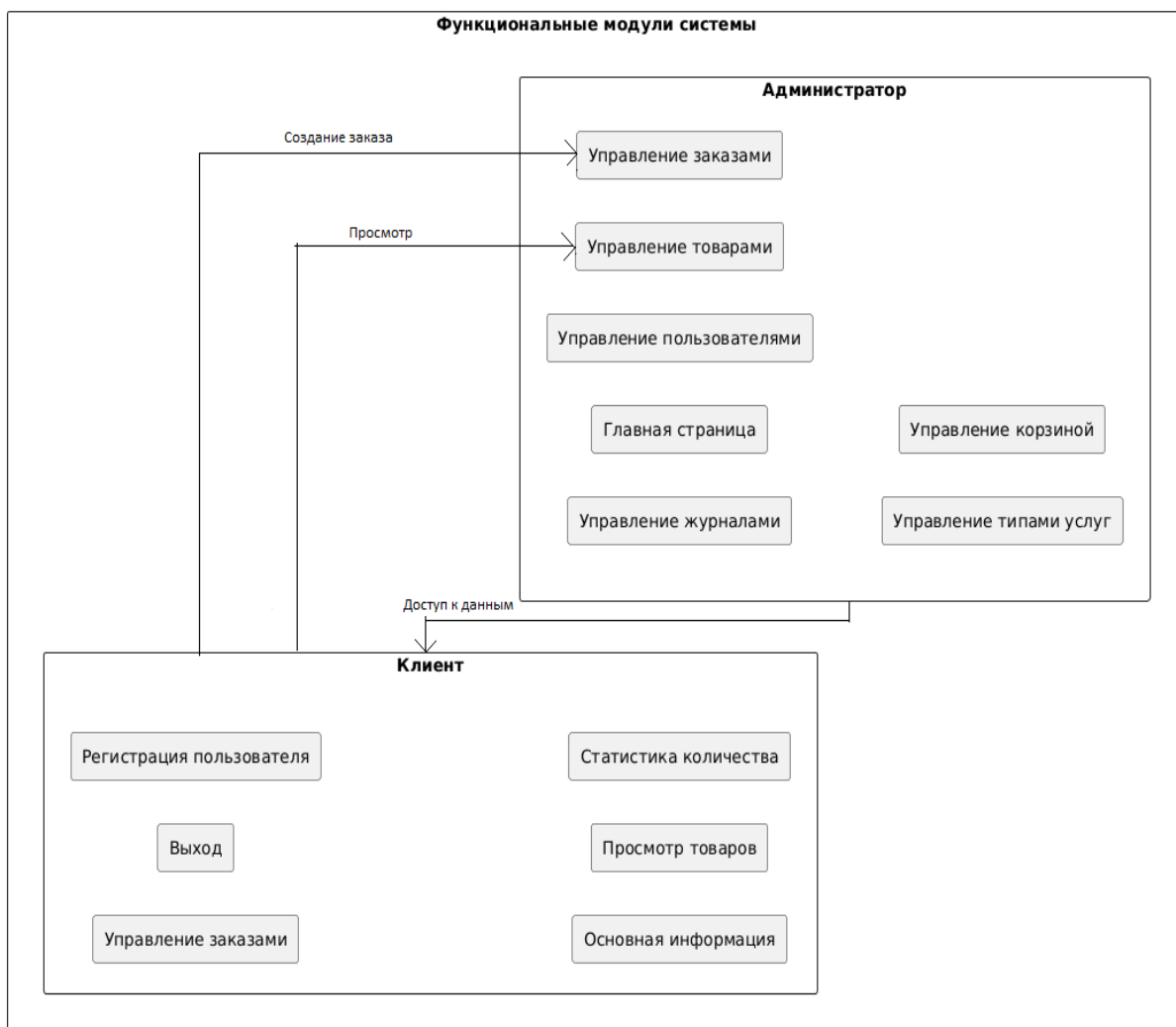


Рисунок 11 – Функциональная структура системы продаж в магазинах цветов и птиц на основе Java

Функциональная структура отдельных модулей системы со стороны администратора:

- администратор размещает веб-страницу, где на главной странице можно посмотреть количество клиентов, количество продавцов, количество заказов за день;
- менеджеры могут добавлять различные виды услуг для продавцов, которые выбирают разные виды добавляемых товаров;
- администратор проверяет информацию о товарах, размещенную

продавцом, для проверки, а товары, прошедшие проверку, могут быть заказаны клиентом или добавлены в корзину;

- администратор просматривает информацию о заказе, размещенном клиентом, просматривает статус обработки заказа через систему журнала.

Функциональная структура системы со стороны клиента:

- клиент может зарегистрировать информацию о пользователе на платформе и просмотреть свою основную информацию после успешной посадки;

- клиент может проверить информацию о товаре продавца, выбрать подходящий товар для размещения заказа или присоединиться к корзине, которая после размещения заказа непосредственно переходит к управлению заказом;

- клиенты могут выйти из десанта на платформе и попасть на лендинг.

Функциональная структура модулей системы со стороны продавца:

- продавец может зарегистрировать информацию о продавце на лендинге, после успешной посадки перейти к основной информации, а продавец может просмотреть личную информацию;

- продавцы могут посмотреть количество заказов за день, пройдя статистику количества;

- продавец создает информацию о товаре путем выбора подходящего типа услуг, предоставляемых администратором, после рассмотрения и прохождения администратором оформления заказа клиентом;

- продавец может проверить выполнение заказа и статус заказа через управление заказом.

Эта система охватывает пользователя, клиента, продавца, администратора, коммерческую организацию, потребителя и заказчика.

Свойства сущности пользователь:

- уникальный идентификатор пользователя (РК);

- имя пользователя;

- пароль;

- номер телефона;

- электронная почта.

Атрибуты сущности роль:

- уникальный идентификатор роли (PK);
- описание роли;
- наименование роли (например: пользователь, продавец, администратор).

Атрибуты сущности товар:

- уникальный идентификатор товара (PK);
- наименование товара;
- цена;
- идентификатор пользователя, добавившего товар;
- код товара (или тип/категория);
- изображение товара.

Атрибуты сущности заказ:

- уникальный идентификатор заказа (PK);
- идентификатор пользователя;
- идентификатор товара.

Атрибуты сущности корзина:

- уникальный идентификатор корзины (PK);
- идентификатор пользователя;
- клиентский ID;
- идентификатор товара.

Атрибуты сущности обратная связь:

- уникальный идентификатор (PK);
- название заказа;
- ID клиента;
- текст отзыва или обращения;
- внешний ключ клиента.

Атрибуты сущности оценка:

- уникальный идентификатор (PK);
- наименование заказа;
- идентификатор клиента;

- оценка (например, текст или числовая);
- внешний ключ клиента.

На рисунке 12 показана схема E-R системы продаж в магазинах цветов и птиц.

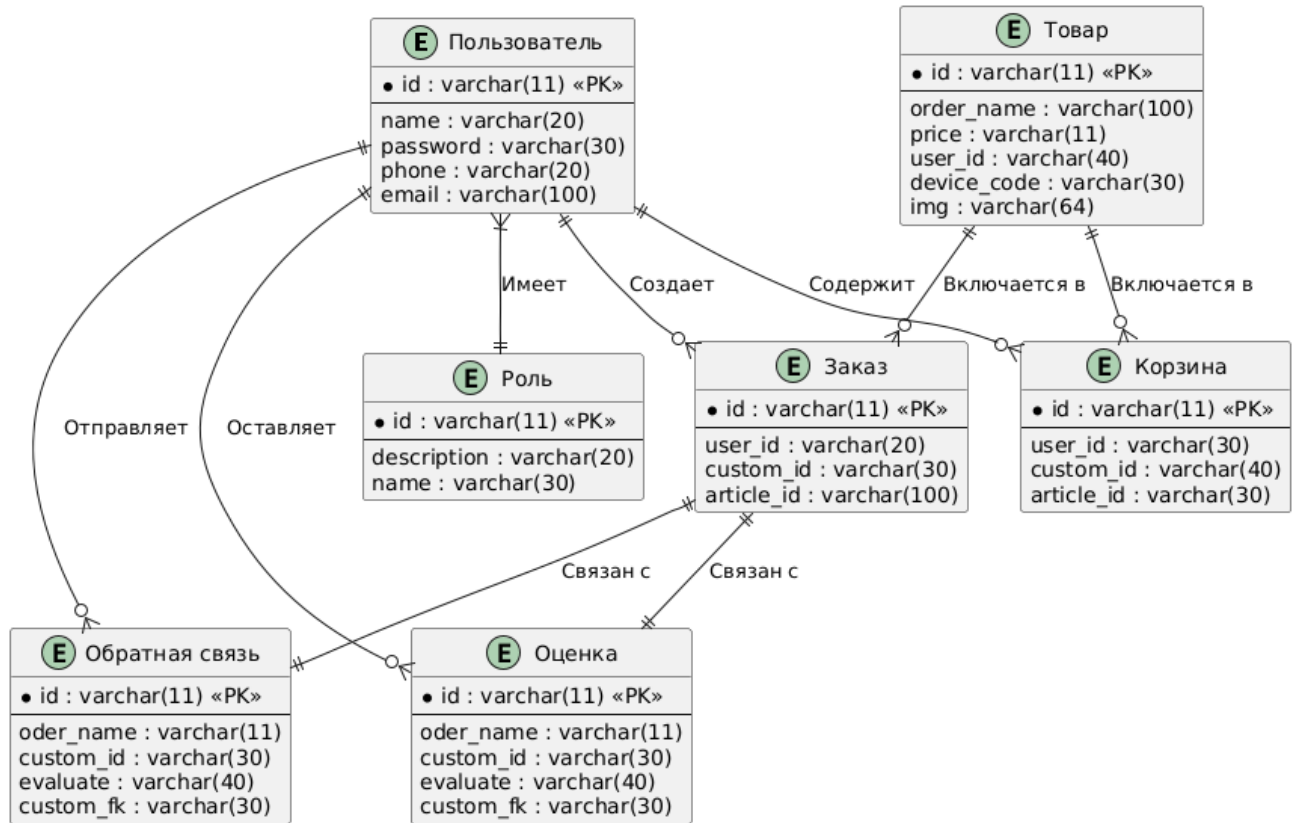


Рисунок 12 – Схема E-R системы продаж в магазине цветов и птиц

На рисунке представлена Entity-Relationship диаграмма базы данных системы, отражающая следующие сущности и их взаимосвязи:

- пользователь сущность, которая содержит данные клиентов (логин, пароль, контакты);
- роль сущность, которая определяет права доступа (администратор, клиент);
- товар сущность, хранящая информацию о продуктах (название, цена, категория);
- корзина временное хранилище товаров перед оформлением заказа;
- заказ сущность, фиксирующая детали покупки (покупатель, товары, статус);
- оценка и обратная связь сущности, позволяющие пользователям

оставлять отзывы.

Связи между сущностями:

- пользователь может иметь одну роль, но создавать множество заказов;
- товары могут быть в нескольких корзинах и заказах;
- каждый заказ может быть связан с оценкой и обратной связью.

Диаграмма создана с использованием стандарта Crow's Foot для обозначения отношений (один-ко-многим).

Благодаря дизайну структуры базы данных для системы продаж магазина цветов и птиц мы можем получить, пользователь, персонаж, товар, корзина, заказ и другие таблицы. Доступ к E-R-схеме базы данных для системы продаж в магазинах цветов и птиц обеспечивается главным образом за счет концептуальной структуры базы данных. Ниже приводится конкретная структура таблицы, полученная в результате обработки диаграмм E-R.

Таблица с данными о пользователях содержит базовую информацию для авторизации и связи с клиентами [14].

В таблице 3 приведена структура таблицы с данными о пользователях.

Таблица 3 – Структура таблицы с данными о пользователях

Имя, соответствующее полю	Значение соответствия поля	Тип, соответствующий полю	ключ
id	идентификатор пользователя	varchar(11)	главный ключ
name	имя пользователя	varchar(20)	
password	пароль	varchar(30)	
фон	Телефон	varchar (20)	
email	почтовый ящик	varchar (100)	

Таблица с информацией о структуре ролей определяет уровень доступа пользователей в системе.

Таблица 4 представляет собой структуру ролей.

Таблица 4 – Структура ролей

Имя, соответствующее полю	Значение соответствия поля	Тип, соответствующий полю	ключ
id	ид персонажа	varchar(11)	главный ключ
d e scription	имя персонажа	varchar(20)	
n ame	имя пользователя	varchar(30)	

Таблица с информацией о товарах включает визуальную информацию о товарах.

В таблицах 5 показана структура таблиц по товарам

Таблица 5 – Структура товарных таблиц

Имя, соответствующее полю	Значение соответствия поля	Тип, соответствующий полю	ключ
id	ид товара	varchar(11)	главный ключ
O der_name	наименование товара	varchar (100)	
price	цена	varchar(11)	
U ser_id	Наименование продавца	varchar(40)	
device code	Тип товара	varchar(30)	
I мГ	картинка	varchar (64)	

Таблица со структурой корзины связывает пользователей с товарами перед оформлением заказа.

В таблице 6 приведена структура таблицы корзины.

Таблица 6 – Структура корзины

Имя, соответствующее полю	Значение соответствия поля	Тип, соответствующий полю	ключ
1	2	3	4
id	идентификатор корзины	varchar(11)	главный ключ
U ser_id	Наименование продавца	varchar(30)	

1	2	3	4
custom_id	Наименование клиента	varchar(40)	
article_id	наименование товара	varchar(30)	

Таблица со структурой заказа фиксирует завершённые покупки и является основной таблицей для анализа продаж.

В таблице 7 приведена структура таблицы заказов.

Таблица 7 – Структура заказа

Имя, соответствующее полю	Значение соответствия поля	Тип, соответствующий полю	ключ
id	идентификатор заказа	varchar(11)	главный ключ
User_id	Наименование продавца	varchar(20)	
custom_id	Наименование клиента	varchar(30)	
article_id	наименование товара	varchar (100)	

Таблица с оценками хранит отзывы о товарах и позволяет собирать feedback от клиентов.

Структура оценочной таблицы показана в таблице 8.

Таблица 8 – Структура оценочной таблицы

Имя, соответствующее полю	Значение соответствия поля	Тип, соответствующий полю	ключ
id	ид оценки	varchar(11)	главный ключ
order_name	№ заказа	varchar(11)	
custom_id	Наименование клиента	varchar(30)	
evaluate	Оценка пользователей	varchar(40)	
custom_fk	Отзывы клиентов	varchar(30)	

Структура таблицы обратной связи используется для технической

поддержки и обрабатывает запросы пользователей.

Структура таблицы обратной связи показана в таблице 9.

Таблица 9 – Структура таблицы обратной связи

Имя, соответствующее полю	Значение соответствия поля	Тип, соответствующий полю	ключ
id	id обратной связи	varchar(11)	главный ключ
oder_name	№ заказа	varchar(11)	
custom_id	Наименование клиента	varchar(30)	
ealuate	Оценка пользователей	varchar(40)	
custom_fk	Отзывы клиентов	varchar(30)	

Представленная структура базы данных охватывает все ключевые аспекты работы интернет-магазина:

- управление пользователями;
- товарооборот;
- аналитика.

Рекомендации по улучшению:

- добавить индексы для часто запрашиваемых полей;
- реализовать мягкое удаление записей;
- внедрить историю изменений цен;
- оптимизировать хранение изображений.

Таким образом, данная схема обеспечивает масштабируемость системы и соответствует современным стандартам проектирования реляционных баз данных.

3.2. Примеры фактического тестирования программного продукта

3.2.1 Структура пользовательского интерфейса

Фон системы доступен только тем пользователям, которые отвечают условиям соответствия. В соответствии с реальными потребностями необходимо проверить правильность и соответствие имени пользователя и пароля пользователя, поэтому был разработан модуль входа в систему продаж магазина цветов и птиц. Для использования фона системы продаж магазина цветов и птиц необходимо,

чтобы имя пользователя и пароль были полностью правильными и соответствовали друг другу.

На рисунке представлена блок-схема процесса аутентификации пользователя в системе. Процесс включает следующие этапы:

- ввод учетных данных ввод логина и пароля пользователем;
- проверка данных сравнение системой введенных данных с информацией в базе данных;
- валидация определение типа пользователя (администратор/клиент) при успешной проверке системой и загрузка соответствующей главной страницы с доступными функциями. При ошибке выводится уведомление, и процесс повторяется до корректного ввода.

На рисунке 13 показана блок-схема входа в систему.

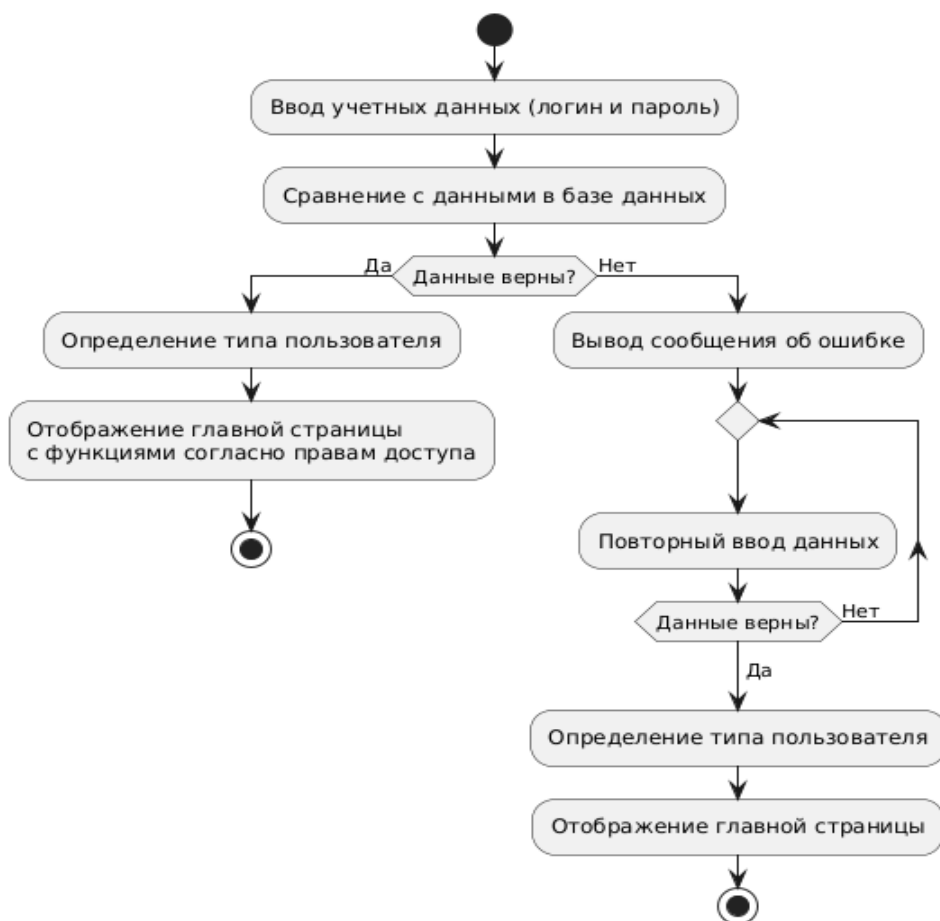


Рисунок 13 – Схема входа в систему продаж магазина

Интерфейс входа в систему отражает цикличность процесса при неверных данных и однократную авторизацию при успешной проверке.

Интерфейс входа в систему показывает основные окна, которые используются в веб-приложении.

Интерфейс входа показан на рисунке 14.

Авторизуйтесь в системе

Имя пользователя

пароль

Вход

Рисунок 14 – Интерфейс входа в систему продаж магазина цветов и птиц

После успешного входа пользователь определяет параметры, которые он может использовать, в зависимости от вида пользователя, установленного администратором, и показывает соответствующий параметр в меню страницы.

На рисунке представлена логика формирования адаптивного интерфейса системы в зависимости от типа пользователя:

- идентификация прав доступа определение типа пользователя (администратор или клиент) системой, после успешной аутентификации через запрос к базе данных;
- динамическая загрузка модулей для администратора отображение всех функции управления, управление товарами и категориями, обработка заказов, администрирование пользователей;
- динамическая загрузка модулей для клиента просмотр каталога товаров, управление корзиной с возможностью оформления заказа, вход в личный кабинет для просмотра истории покупок;
- визуализация меню скрытие системой недоступных для текущего типа пользователя модулей тем самым обеспечивая безопасность и удобство интерфейса.

Блок-схема программы в меню показана на рисунке 15.



Рисунок 15 – Схема программы в меню

Эта функция предназначена главным образом для модерации товаров пользователями, категория пользователей которых является администратором.

Интерфейс динамического меню показан на рисунке 16.

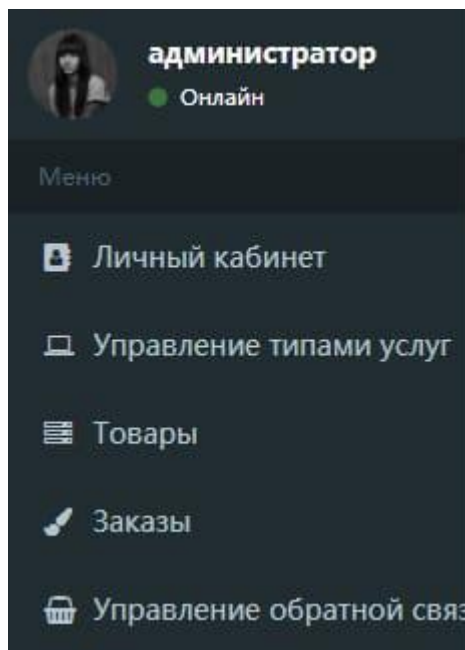


Рисунок 16 – Граница меню

На рисунке представлен процесс модерации товаров в системе:

- загрузка данных – запрос от системы из базы данных полного списка товаров, ожидающих проверки. Информация отображается в интерфейсе администратора;
- проверка товара – анализ администратором корректности описания и изображений, проверка соответствие категории и адекватности цены;
- результат проверки – получение товаром статуса «Одобрено». При успешной проверке товар становится видимым для покупателей. При отклонении указываются причины и товар возвращается продавцу на доработку.

На рисунке 17 показана блок-схема проверки товаров администратором.

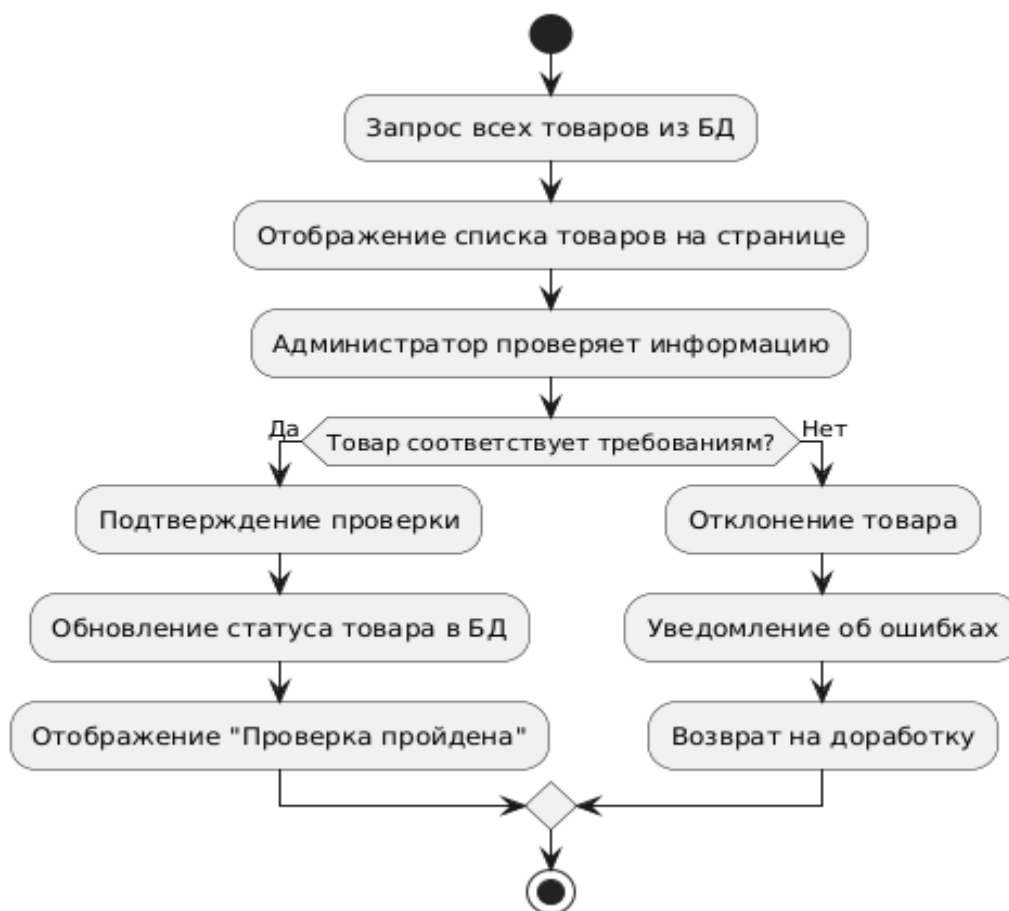


Рисунок 17 – Схема программы рассмотрения товара администратором

Эта функция предназначена главным образом для аудита заказов пользователями, категория пользователей которых является администратором.

Функциональные возможности:

- принятие товара;
- сортировка по колонкам;
- поиск через заголовок таблицы.

Визуальные элементы:

- иконка изображения товара;
- цветовое кодирование статусов.

Загрузка заказов:

- система получает все текущие заказы из базы данных.

Администратор проверяет:

- наличие товаров на складе;

- корректность данных клиента;
- соответствие суммы оплаты;
- доступность выбранного способа доставки.

При подтверждении результата проверки:

- система резервирует товары;
- клиент получает уведомление о подтверждении;
- заказ передается в логистическую систему.

При отклонении:

- клиент получает развернутое объяснение.

При необходимости инициируется процесс возврата средств

Блок-схема проверки заказа администратором показана на рисунке 18.



Рисунок 18 – Блок-схема рассмотрения заказа администратором

Тип интерфейса:

- панель управления заказами для администратора.

Разделы интерфейса:

- навигационная цепочка (главная, управление заказами).

Управляющие кнопки:

- список – перезагрузка данных;
- обновить – синхронизация с БД.

Интерфейс для управления заказами администратором показан на рисунке

19.

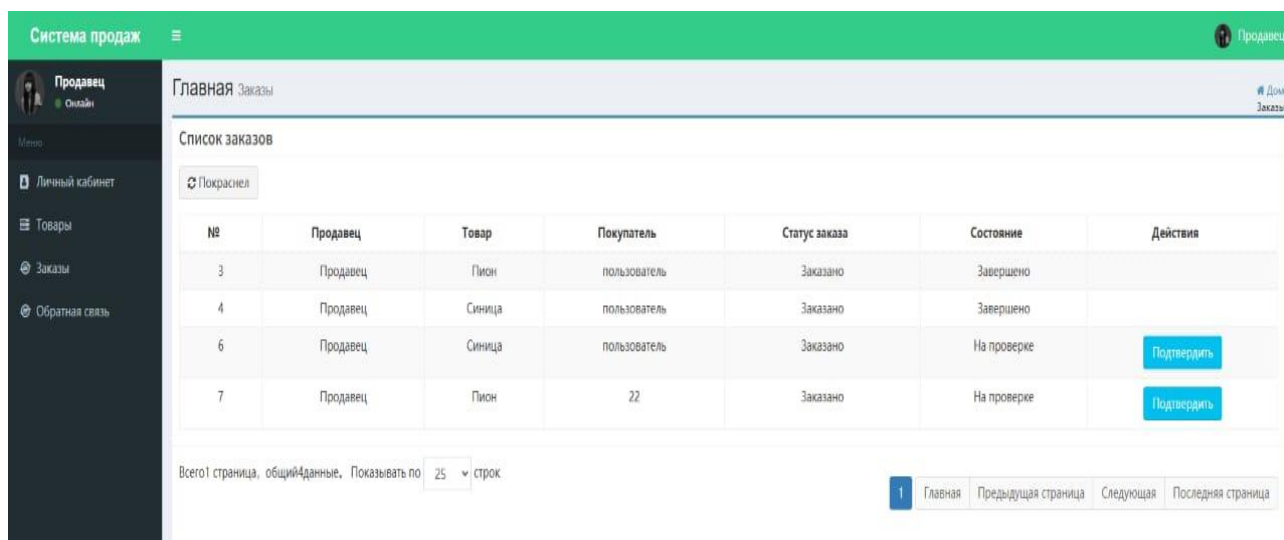


Рисунок 19 – Интерфейс для управления заказами администратором

Данная функция в основном является добавлением информации о товаре для пользователей, категория пользователей которых является продавцами.

На схеме представлен алгоритм добавления товарных позиций в систему.

Авторизация:

- доступ к функции предоставляется только аутентифицированным продавцам;

- проверка роли пользователя (роль Продавец).

Обязательные поля процесса добавления:

- наименование товара (макс. 100 символов);
- категория (выбор из справочника);
- цена (числовой формат);
- минимум 1 изображение.

Опциональные поля:

- подробное описание;
- характеристики;
- срок годности (для определенных категорий).

Схема добавления информации о новых товарах продавцом показана на рисунке 20.

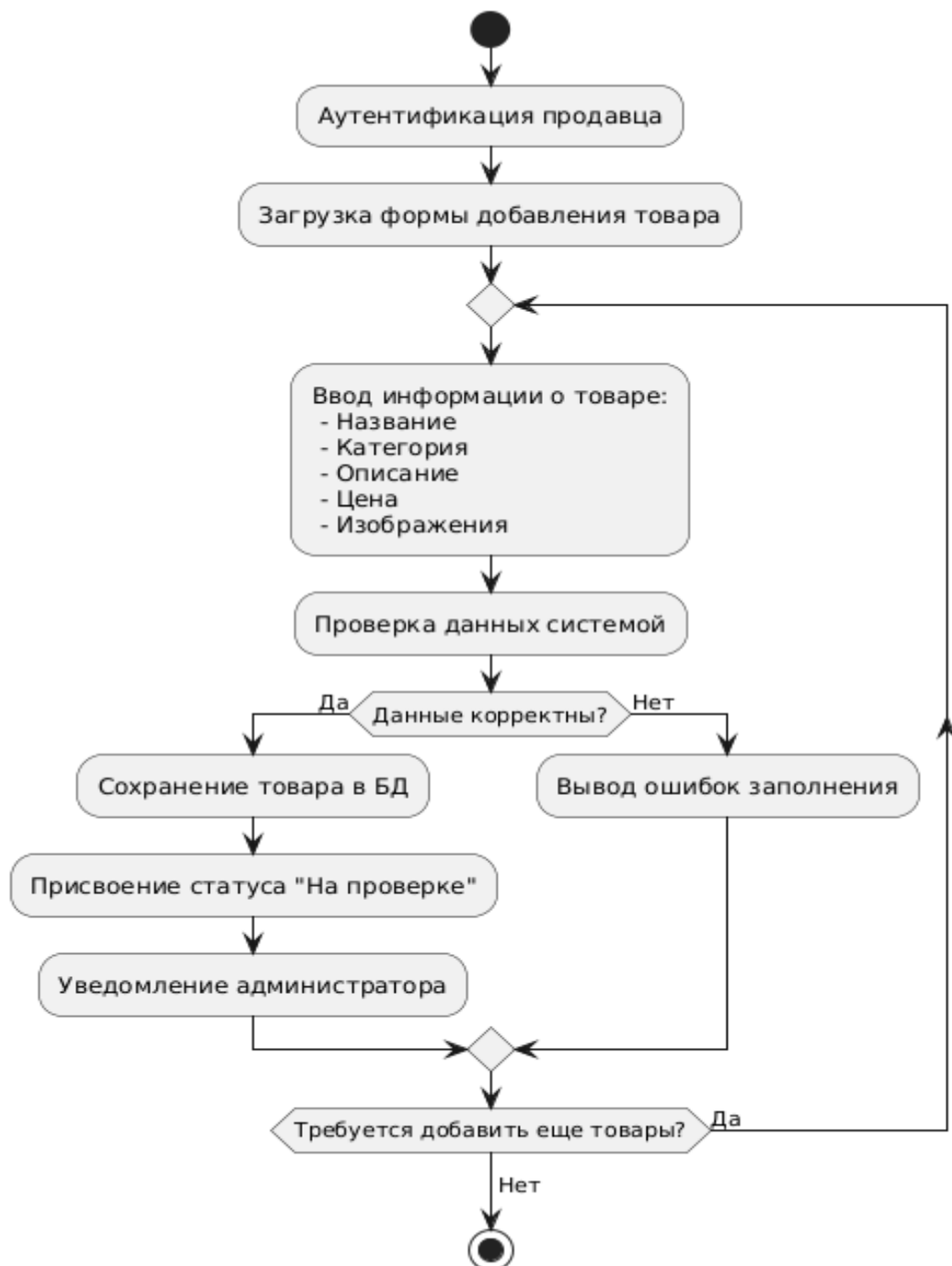


Рисунок 20 – Схема информации о новых товарах, добавляемых продавцом

Тип интерфейса:

- форма добавления нового товара для продавца.

Разделы интерфейса:

- навигация (управление товарами, добавление товара);
- указание текущего действия (информация о товаре).

Схема интерфейса для дополнительной информации о товаре, представленной продавцом, показана на рисунке 21.

Рисунок 21 – Схема интерфейса информации о новых товарах, добавленных продавцом

Данная функция в основном представляет собой аудит информации о заказе для пользователей, категория пользователей которых является продавцами.

На схеме представлен полный цикл работы продавца с заказами.

Авторизация и фильтрация:

- система отображает только заказы, содержащие товары данного продавца;
- используется фильтрация по статусам («Новые», «Ожидают обработки»).

Дополнительные проверки:

- ограничения по региону доставки;
- специальные условия покупателя.

Варианты решений:

- подтверждение и резервирование товара;
- отклонение с обязательным указанием причины (нет в наличии, изменилась цена, ошибка в заказе).

Схема рассмотрения информации о заказе продавцом показана на рисунке 22.

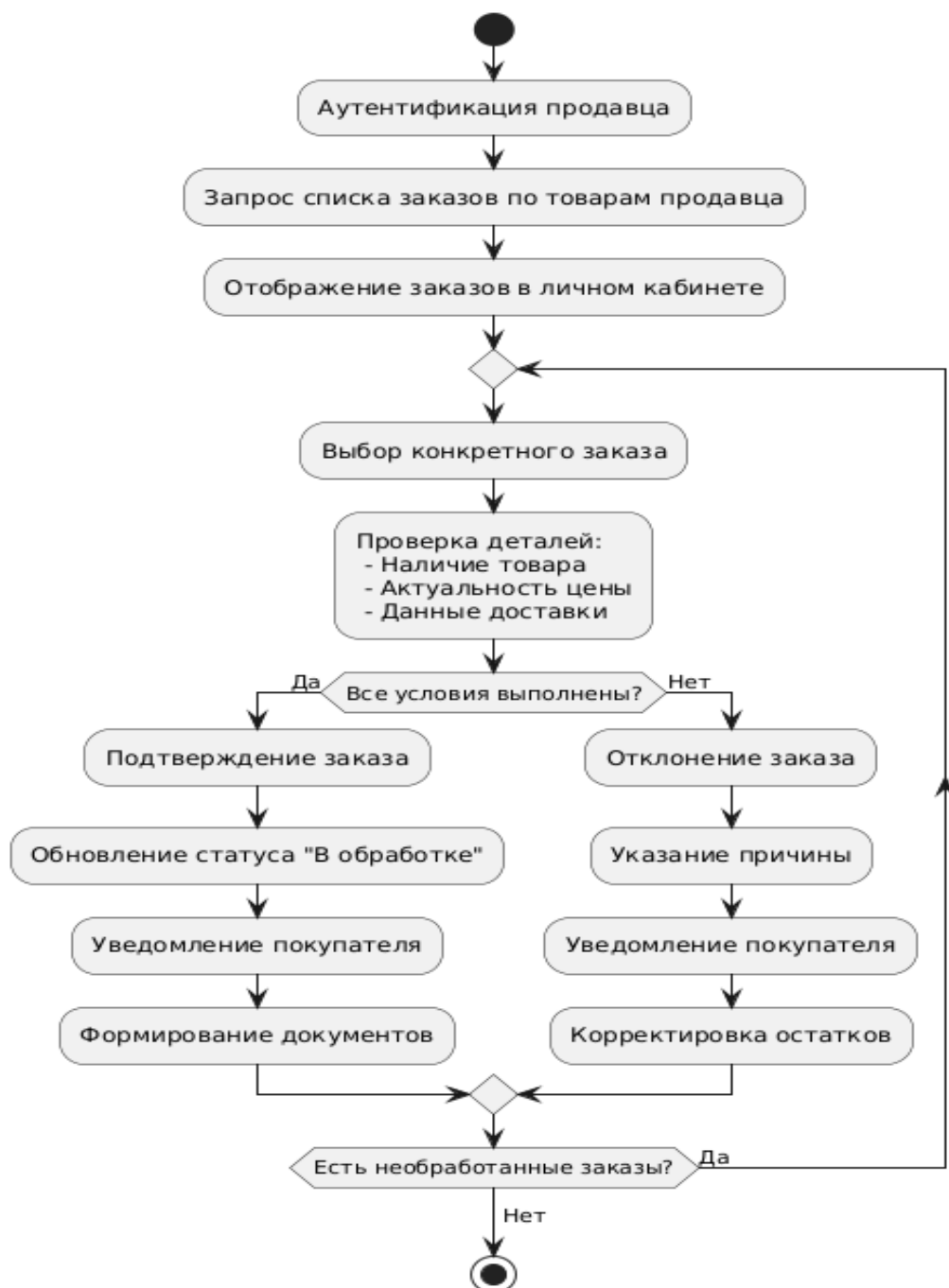


Рисунок 22 – Схема информации о рассмотрении заказа продавцом

Схема интерфейса для рассмотрения информации о заказе продавцом показана на рисунке 23.

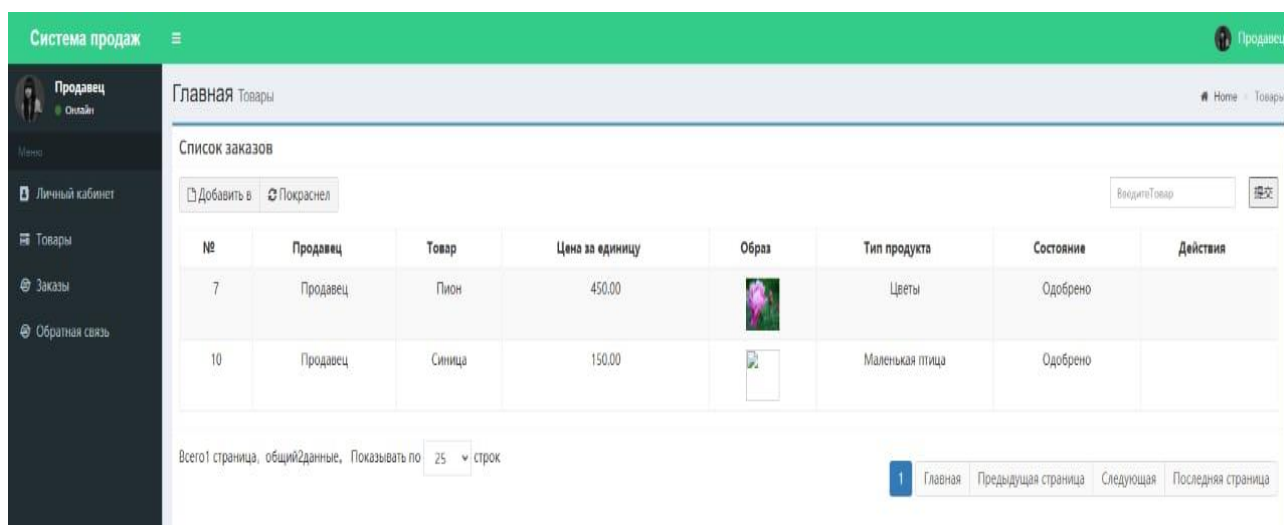


Рисунок 23 – Схема интерфейса информации о заказе, рассмотренном продавцом

Эта функция в основном добавляется к корзине для пользователей, категория пользователей которых является клиентом.

На схеме представлен алгоритм работы клиента с корзиной покупок.

Предварительные условия:

- требуется авторизация (кроме гостевого режима);
- доступ к актуальному каталогу товаров.

Особенности реализации:

- реактивное обновление суммы;
- временное резервирование товара.

Кросс-секционные проверки:

- актуальность цены;
- ограничения по региону.

Блок-схема добавления корзины для клиента показана на рисунке 24.

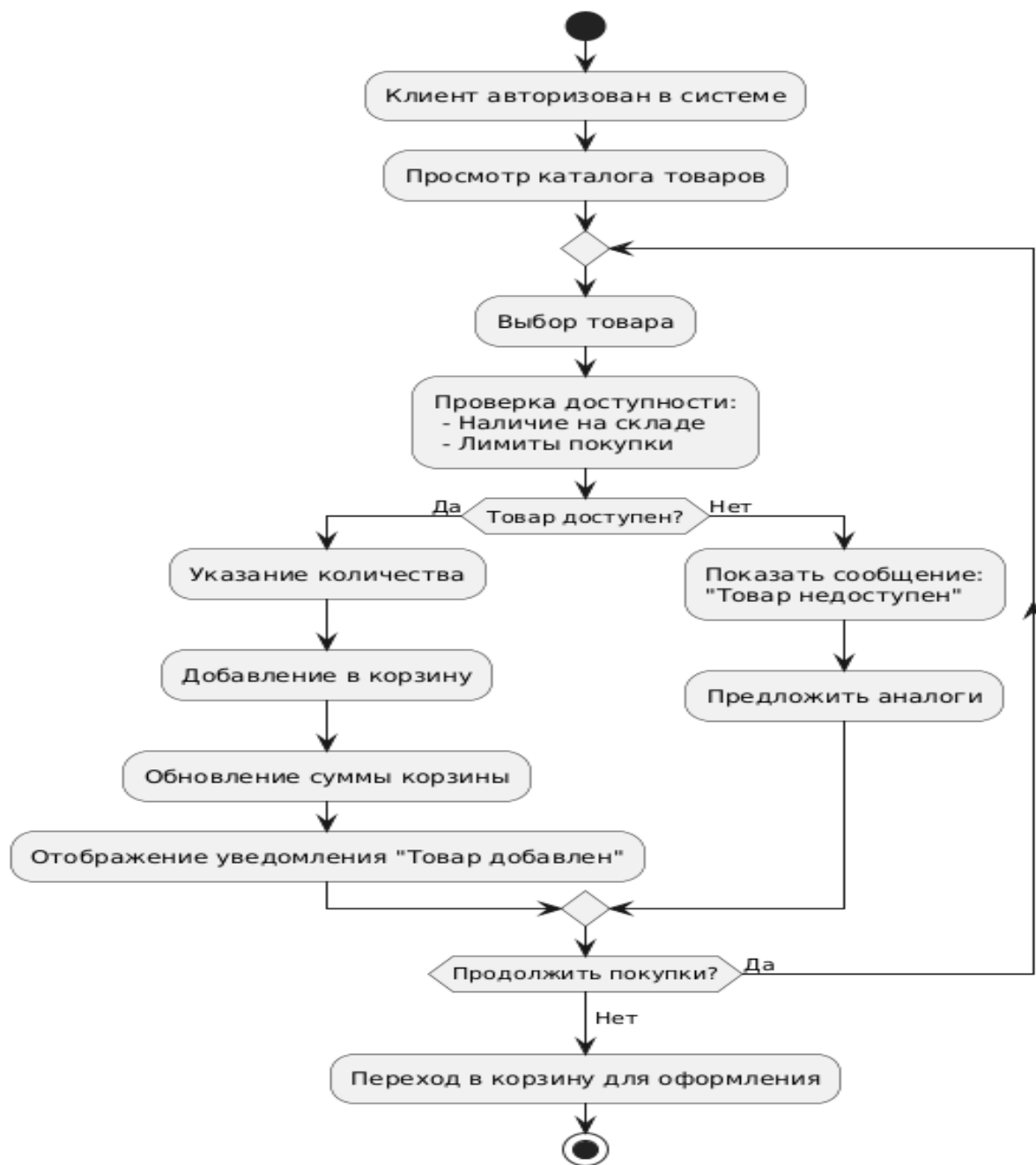


Рисунок 24 – Процесс добавления товаров в корзину клиентом

Интерфейс добавления корзины для клиента показан на рисунке 25.

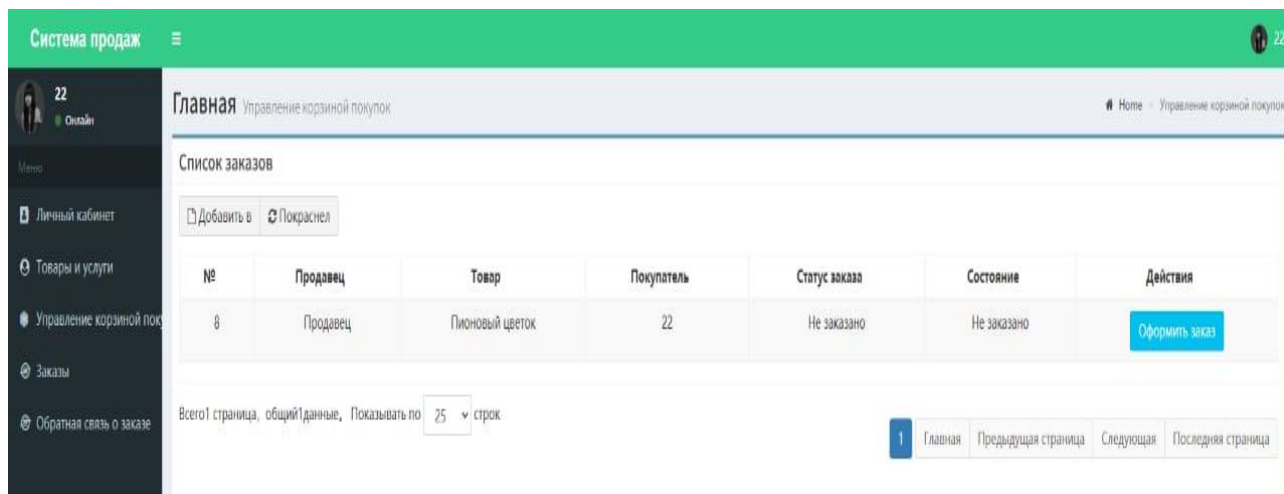


Рисунок 25 – Интерфейс добавленной корзины клиента

Данная функция в основном является добавлением к заказу от пользователей, категория пользователей которых является клиентом.

На схеме представлен полный цикл создания заказа.

Инициализация процесса:

- доступ только для авторизованных клиентов;
- интеграция с корзиной покупок.

Критические проверки:

- наличие товаров на складе;
- валидность адреса доставки;
- корректность контактных данных.

Финансовые операции:

- расчет итоговой суммы с учетом скидок, стоимости доставки, налогов.

Уведомления:

- клиенту на подтверждение заказа;
- продавцу на новый заказ;
- складу на резервирование товаров.

Схема добавления новых заказов для клиента, как показано на рисунке 26.



Рисунок 26 – Схема новых заказов заказчика

График интерфейса для новых заказов клиента демонстрирует все оставленные клиентом заказы, которые необходимо обработать.

График интерфейса для новых заказов клиента показан на рисунке 27.

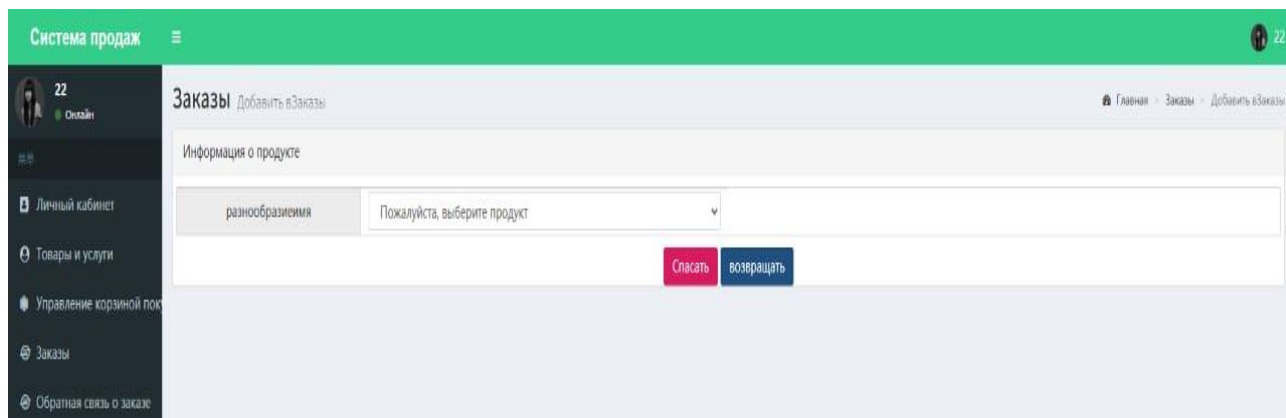


Рисунок 27 – Схема интерфейса нового заказа клиента

3.2.2 Тестирование пользовательского интерфейса

Испытание систем является важнейшим звеном в процессе разработки программного обеспечения, цель которого заключается в оценке разработанных систем программного обеспечения для проверки их производительности и функциональности [5]. Основная цель испытаний системы заключается в обнаружении и оценке нормальности функционирования системы программного обеспечения во всех аспектах, таких как интерфейс, производительность, стабильность и безопасность, а также в выявлении и устранении недостатков и лазеек, улучшении качества и надежности системы программного обеспечения.

Процесс тестирования системы включает в себя несколько этапов планирования, осуществления, анализа и представления отчетности. На этапе планирования тестировщикам необходимо определить тест-кейсы, тестовую среду, инструменты тестирования и другие элементы в соответствии с требованиями спецификации требований к программному обеспечению. На этапе выполнения тестировщик выполняет тест-кейсы в соответствии с планом тестирования и записывает результаты выполнения и любые ошибки или отклонения. На этапе анализа тестировщики проанализируют результаты тестов, определяют тип, тяжесть и влияние дефектов и уязвимостей, а также поделится этой информацией с разработчиками, чтобы она была исправлена. На этапе отчетности испытатель составляет подробный отчет о проведенных испытаниях, а также о результатах испытаний и представляет его членам команды и руководству.

Роль системного тестирования незаменима, и тестировщики играют в нем чрезвычайно важную роль [9]. Благодаря системному тестированию можно все-сторонне оценить производительность, функциональность, безопасность и надежность программного комплекса, выявить и устранить все возможные недостатки и лазейки на раннем этапе, тем самым повысив качество и надежность программного комплекса и обеспечив его способность эффективно обслуживать пользователей.

Динамическое меню в основном отображает различные функции после посадки, согласно тестовым вариантам, результаты теста получены, как показано в таблице 10.

Таблица 10 – Вариант тестирования динамического меню

шаг испытания	входные данные	Ожидаемые результаты	Полученные результаты
Введите пароль аккаунта администратора в интерфейсе посадки	Номер счета: admin Пароль: admin	Функциональное соответствие успешно	Как и ожидалось
Введите пароль счета клиента в интерфейсе посадки	Номер счета: Бизнес Пароль: 123456	Функциональное соответствие успешно	Как и ожидалось
Введите пароль аккаунта продавца в интерфейсе посадки	Аккаунт: user Пароль: 123456	Функциональное соответствие успешно	Как и ожидалось

Проверка товара администратором в основном заключается в том, что администратор проверяет прохождение проверки товара, проверяет функцию товара администратором, в зависимости от варианта использования, полученные результаты испытаний приведены в таблице 11.

Таблица 11 – Варианты тестирования товаров, проверенных администратором

шаг испытания	входные данные	Ожидаемые результаты	Полученные результаты
1	2	3	4
На странице модерации товаров администратором	Нажмите на проход	Статус перешел в режим пройденный	Связь с ожидаемыми результатами одинаковый

1	2	3	4
Нажмите на страницу, где товары рассматриваются администратором, и не пройдите	Нажмите для определения	превратить положение в непройденное	Связь с ожидаемыми результатами одинаковый

Проверка заказа администратором в основном заключается в прохождении аудита заказа администратором, проверка функции заказа администратором по варианту использования, результаты испытаний, полученные в соответствии с вариантом использования, приведены в таблице 12.

Таблица 12 – Варианты тестирования заказов, рассмотренных администратором

шаг испытания	входные данные	Ожидаемые результаты	Полученные результаты
На странице аудита заказов администратором	Нажмите на проход	Статус перешел в режим пройденный	Связь с ожидаемыми результатами одинаковый
Нажатие на верхнюю страницу аудита администратора не проходит	Нажмите для определения	превратить положение в непройденное	Связь с ожидаемыми результатами единогласие

Добавление информации о товаре продавцом в основном является добавлением информации о товаре пользователем, который устанавливает вид пользователя продавца, в зависимости от варианта использования.

В таблице 13 показаны тестовые варианты информации о новых товарах, добавляемых продавцами.

Таблица 13 – Тестовые варианты информации о новых товарах, добавляемых продавцами

шаг испытания	входные данные	Ожидаемые результаты	Полученные результаты
Нажмите на новый товар	Название сорта: Цветочный продукт Имя: Лилия Цена за единицу: 10 Фото: baihe.jpg	успех	Связь с ожидаемыми результатами одинаковый
Нажмите на новый товар	Нажмите на отмену	успех	Связь с ожидаемыми результатами одинаковый

Испытание модуля информации о заказе на аудит продавца в основном заключается в прохождении аудита заказа администратором, в соответствии с вариантом использования, как показано в таблице 14.

Таблица 14 – Вариант тестирования информации о заказе на рассмотрение продавца

шаг испытания	входные данные	Ожидаемые результаты	Полученные результаты
Нажмите на страницу оформления заказа администратором, чтобы пройти	Нажмите для определения	Статус перешел в режим пройденный	Связь с ожидаемыми результатами одинаковый
Нажатие на верхнюю страницу аудита администратора не проходит	Нажмите для определения	превратить положение в непройденное	Связь с ожидаемыми результатами одинаковый

Тестирование модуля служебной информации с обратной связью продавца в основном является обратной связью продавца с оценкой пользователя, тестом служебной информации с обратной связью продавца по варианту использования.

В таблице 15 показан вариант тестирования информации о заказе на

рассмотрение продавца.

Таблица 15 – Вариант тестирования информации о заказе на рассмотрение продавца

шаг испытания	входные данные	Ожидаемые результаты	Полученные результаты
Нажмите на страницу информации о сервисе обратной связи с продавцом для оценки	Большое спасибо	успех	Связь с ожидаемыми результатами одинаковый
Нажмите на страницу оценки информации о сервисе обратной связи продавца, чтобы отменить	Нажмите на отмену	успех	Связь с ожидаемыми результатами одинаковый

Тестирование модуля добавления корзины клиентом в основном заключается в добавлении товаров клиентом в корзину, добавлении теста корзины для клиента, согласно варианту использования, как показано в таблице 16.

Таблица 16 – Тестовый вариант добавления корзины клиентом

шаг испытания	входные данные	Ожидаемые результаты	Полученные результаты
Добавить корзину на страницу клиента Нажмите добавить	Выберите товар, нажмите на сохранение	успех	Связь с ожидаемыми результатами единогласие
Добавить страницу в корзину клиента	Нажмите на отмену	успех	Связь с ожидаемыми результатами одинаковый

Заказчик добавляет модуль заказа. Тестирование в основном заключается в том, что клиент добавляет товар в заказ и добавляет модуль заказа к клиенту, что в зависимости от варианта использования.

В таблице 17 показаны варианты тестирования модуля добавления заказов для клиентов.

Таблица 17 – Варианты тестирования модуля добавления заказов для клиентов

шаг испытания	входные данные	Ожидаемые результаты	Полученные результаты
Добавить заказ на страницу клиента Нажмите Добавить	Выберите товар, нажмите на сохранение	успех	Связь с ожидаемыми результатами одинаковый
Добавить заказ на страницу клиента Нажмите Добавить	Нажмите на отмену	успех	Связь с ожидаемыми результатами одинаковый

Тестирование модуля оценки заказа клиентом в основном заключается в оценке клиентом выполненного заказа, а модуля оценки заказа клиентом в зависимости от варианта использования, как показано в таблице 18.

Таблица 18 – Тестовый вариант оценки заказа клиентом

шаг испытания	входные данные	Ожидаемые результаты	Полученные результаты
Нажмите на страницу оценки заказа клиентом Оценка	Оценка: Товар очень качественный	успех	Связь с ожидаемыми результатами одинаковый
Нажмите на страницу оценки заказа клиентом Оценка	Нажмите на отмену	успех	Связь с ожидаемыми результатами одинаковый

В результате тестирования модуля продаж в магазине цветов и птиц были выявлены проблемы с безопасностью паролей пользователей в системе. Использование фреймворка Shiro для обработки пароля шифрованием позволило решить проблему безопасности, и теперь система работает стабильно. Этот процесс разработки и тестирования оказал важную помощь в совершенствовании разработки и сочетании теоретических знаний и практики, а также проверил полученные знания.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной выпускной квалификационной работы была успешно достигнута поставленная цель разработка и реализация информационной системы управления продажами для цветочных и зоомагазинов с использованием языка программирования Java. Выполненная работа продемонстрировала возможности современных технологий программирования для автоматизации процессов розничной торговли и повышения эффективности бизнес-деятельности.

Проект охватывает ключевые аспекты функционирования торговой точки: от ведения базы товаров и оформления заказов до учёта клиентов и формирования отчетов. В процессе разработки были использованы подходы объектно-ориентированного проектирования, построены UML-диаграммы, реализованы функциональные модули с графическим интерфейсом, а также проведено тестирование, подтверждающее корректность работы приложения.

Созданная система обеспечивает удобную и логичную организацию пользовательского взаимодействия, позволяет сократить время обработки заказов, исключить ошибки, связанные с ручным учетом, и повысить прозрачность бизнес-процессов. Применение Java обеспечило кроссплатформенность и возможность масштабирования решения под различные типы магазинов.

Результаты данной работы обладают высокой прикладной ценностью и могут быть использованы для цифровизации малого и среднего бизнеса в сфере розничной торговли. Разработанное программное обеспечение может быть внедрено в действующие магазины или использовано как основа для дальнейшего развития в рамках коммерческих или учебных проектов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Абрамов, С. Н. Проектирование человеко-машинных интерфейсов: учебное пособие / С. Н. Абрамов. – М.: Горячая линия – Телеком, 2021. – 240 с.
- 2 Беляев, В. Ю. Информационные технологии в торговле и логистике / В. Ю. Беляев. – М.: Академический проект, 2021. – 294 с.
- 3 Бочаров, А. А. Проектирование информационных систем: учебное пособие / А. А. Бочаров. – М.: КноРус, 2022. – 256 с.
- 4 Ван Лян. Модульная разработка интернет-магазина на Java EE // Электронные вычисления и технологии. – 2021. – №10.
- 5 Вэй Мин. Безопасность данных в web-приложениях на Java: методы защиты // Информационная безопасность. – 2022. – №9.
- 6 Гао Юн. Разработка и реализация системы бронирования товаров на Java // Компьютерная инженерия и автоматизация. – 2020. – №9.
- 7 Глушаков, И. В. Java. Объектно-ориентированное программирование: учебное пособие / И. В. Глушаков. – М.: Диалог-МИФИ, 2019. – 296 с.
- 8 Го Хуэй. Автоматизация бизнес-процессов с помощью Java и Spring Boot // Цифровая трансформация. – 2023. – №2.
- 9 Гордон, А. И. Разработка баз данных и интерфейсов пользователя: учебное пособие / А. И. Гордон. – СПб.: Питер, 2020. – 288 с.
- 10 Ермаков, С. С. Программная инженерия: основы и практика: учебное пособие / С. С. Ермаков. – М.: Инфра-М, 2022. – 384 с.
- 11 Журавлёв, С. В. Разработка программного обеспечения: методология, проектирование, технология / С. В. Журавлёв. – М.: Форум, 2020. – 320 с.
- 12 Карташов, И. В. Информационные системы: проектирование и внедрение / И. В. Карташов. – СПб.: Питер, 2019. – 336 с.
- 13 Козлов, И. Б. Java на практике: учебное пособие для вузов / И. Б. Козлов. – М.: Бином. Лаборатория знаний, 2021. – 414 с.
- 14 Корольков, А. А. Технологии программирования в среде Java / А. А. Корольков. – М.: Форум, 2022. – 312 с.

- 15 Кузнецов, С. В. Архитектура программных систем: учебное пособие / С. В. Кузнецов. – СПб.: Питер, 2020. – 328 с.
- 16 Лаптев, А. А. Технологии программирования: учебник / А. А. Лаптев. – СПб.: Питер, 2021. – 368 с.
- 17 Ли Мэн. Разработка системы учёта клиентов цветочного магазина на Java // Информационные технологии и приложения. – 2021. – №5.
- 18 Линь Нань. Проектирование системы управления доставкой товаров с использованием Java // Транспорт и автоматизация. – 2021. – №10.
- 19 Литвинов, К. А. Разработка приложений на Java с использованием баз данных / К. А. Литвинов. – М.: Финансы и статистика, 2020. – 342 с.
- 20 Лю Вэй. Проектирование и реализация информационной системы розничной торговли на Java // Современные технологии. – 2021. – №9.
- 21 Лю Хай. Использование AJAX в разработке динамических Java Web-интерфейсов // Электронные вычислительные системы. – 2021. – №11.
- 22 Ма Юэ. Исследование UI/UX-дизайна в системах управления на Java Web // Интерфейсы и взаимодействие. – 2022. – №8.
- 23 Малыхина, М. П. Базы данных: основы, проектирование, использование / М. П. Малыхина. – СПб.: BHV, 2019. – 528 с.
- 24 Николаев, С. П. Java-программирование: основы и практика / С. П. Николаев. – М.: Бином. Лаборатория знаний, 2023. – 356 с.
- 25 Орлов, А. М. Системный анализ и проектирование программного обеспечения / А. М. Орлов. – М.: КноРус, 2022. – 304 с.
- 26 Пирогов, В. Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В. Ю. Пирогов. – СПб.: БХВ-Петербург, 2021. – 440 с.
- 27 Селезнёв, Д. А. Основы объектно-ориентированного анализа и проектирования: учебное пособие / Д. А. Селезнёв. – СПб.: БХВ-Петербург, 2020. – 272 с.
- 28 Стружкин, Н. П. Базы данных: проектирование. практикум: Учебное пособие / Н. П. Стружкин, В. В. Годин. – Люберцы: Юрайт, 2020. – 291 с.

- 29 Тан Цзянь. Использование шаблонов проектирования в Java-приложениях // Программная инженерия. – 2020. – №12.
- 30 Тимофеев, А. С. Разработка и сопровождение программных продуктов: учебник / А. С. Тимофеев. – М.: Юрайт, 2020. – 348 с.
- 31 Титов, В. А. Информационные технологии в торговле и бизнесе: учебное пособие / В. А. Титов. – М.: Академия, 2019. – 275 с.
- 32 Фёдоров, Н. Г. Основы проектирования интерфейсов пользователя / Н. Г. Фёдоров. – М.: Инфра-М, 2019. – 198 с.
- 33 Ху Мин. Интеграция фронтенд и бэкенд частей в Java Web-системах // Веб-технологии. – 2022. – №5.
- 34 Цзинь Чжун. Реализация админ-панели для системы продаж на Java // Вестник цифровых решений. – 2022. – №6.
- 35 Чжан Лэй. Java Web-приложения и обработка запросов с помощью Servlet и JSP // Программирование и информационные технологии. – 2023. – №2.
- 36 Чжан Цзюнь. Разработка и реализация корпоративной системы управления людскими ресурсами на основе Java // Информатизация Китая. – 2022. – №3.
- 37 Чжэн Шаосюн. Проектирование и разработка корпоративной системы управления запасами на основе Java // Инфокоммуникация. – 2020. – №9.
- 38 Чэнь Хао. Анализ и реализация платформы онлайн-заказов на Java Web // Электронные системы и ПО. – 2021. – №7.
- 39 Яковлев, А. В. Современные методы проектирования программных систем / А. В. Яковлев. – М.: Солон-Пресс, 2021. – 310 с.
- 40 Ян Чжихао. Решение проблемы переходов страниц в Java Web Programming // Electronic Production. – 2020. – №20.
- 41 Амурский государственный университет [Электронный ресурс]. – Режим доступа: <https://amursu.ru/>. – 03.03.2025.
- 42 CodinGame [Электронный ресурс]. – Режим доступа: <https://codingame.com/>. – 15.04.2025.
- 43 JavaRush [Электронный ресурс]. – Режим доступа: <https://javarush.ru/>. – 22.02.2025.