

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: Проектирование и реализация информационной системы управления
продуктовым магазином

Исполнитель
студент группы 3105-ом2

(подпись, дата)

Лю Инхао

Руководитель
доцент, канд. техн. наук

(подпись, дата)

О.В. Жилиндина

Руководитель научного
содержания программы
магистратуры
профессор, доктор техн. наук

(подпись, дата)

И.Е. Ерёмин

Нормоконтроль
инженер кафедры

(подпись, дата)

В.Н. Адаменко

Рецензент
доцент, канд. техн. наук

(подпись, дата)

Л.В. Никифорова

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

УТВЕРЖДАЮ

Зав. кафедрой

_____ А.В. Бушманов

« ____ » _____ 2025 г.

З А Д А Н И Е

К магистерской диссертации студента группы 3105-ом2 _____ Лю Инхао _____

1. Тема магистерской диссертации: Проектирование и реализация информаци-
онной системы управления продуктовым магазином

(Утверждено приказом от 06.03.2025 № 609-уч)

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

3. Исходные данные к магистерской диссертации: аналитико-статистические
данные по бизнес-процессам, каталог используемых разработок, нормативные
и научные источники, данные из открытых интернет-ресурсов

4. Содержание магистерской диссертации (перечень подлежащих разработке
вопросов): разработка замкнутого цикла управления цепочкой поставок, пла-
нирование закупок, взаимодействие с поставщиками, выполнение заказов,
контроль качества, складскую логистику и финансовые расчёты

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

6. Рецензент магистерской диссертации: Л.В. Никифорова, доцент, канд. техн.
наук

7. Дата выдачи задания 14.01.2025

Руководитель выпускной квалификационной работы: _____

О.В. Жилиндина, доцент, канд. техн. наук

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

Заявление принял к исполнению _____

РЕФЕРАТ

Магистерская диссертация содержит 80 страниц, 28 рисунков, 9 таблиц, 31 источник

СИСТЕМА УПРАВЛЕНИЯ, ПРОДУКТОВЫЙ МАГАЗИН, УПРАВЛЕНИЕ ПРОДАЖАМИ, SPRINGBOOT

С популяризацией компьютеров все сферы жизни неотделимы от поддержки компьютеров, и крупные и средние супермаркеты не являются исключением. Некоторые небольшие продуктовые магазины также используют компьютеры для управления. Однако метод управления продуктами магазинами по-прежнему основан на ручном управлении и является основным методом, при котором большое количество заказов и различной информации хранится в бумажных документах и бланках, которые не могут обновляться и проверяться в реальном времени, что не способствует длительному хранению. Существует острая необходимость в дешевой и полнофункциональной системе управления продуктовым магазином, которая поможет менеджерам продуктовых магазинов в их работе. Система управления продуктовым магазином сочетает в себе преимущества гуманизации, высокой эффективности и удобства. Данная система повышает эффективность работы и уменьшает ошибки и упущения в хранении данных за счет научных методов управления и удобных сервисов. работе описывается процесс внедрения системы управления продуктовым магазином, объясняется анализ функциональных требований к системе, а также описывается детальное проектирование и реализация системы.

СОДЕРЖАНИЕ

Введение	5
1 Общая характеристика исследуемой задачи	6
1.1 Описание процессов по управлению продуктовым магазином	6
1.2 Обзор существующих методов решения аналогичных задач	14
2 Алгоритмические и программные решения по управлению магазином	24
2.1 Модульная структура приложения и его функциональные зависимости	24
2.2 Программные платформы для разработки	37
3 Программная реализация предлагаемого алгоритма решения задачи	43
3.1 Основные этапы практической разработки программного продукта	43
3.2 Примеры фактического тестирования программного продукта	68
Заключение	74
Библиографический список	76

ВВЕДЕНИЕ

С развитием экономики на городских и сельских улицах и в общинах появилось множество супермаркетов. Хотя малые и средние супермаркеты становятся все более распространенными, в их методах управления по-прежнему преобладает ручное управление. Большое количество заказов на товары и генерируемая информация о поставщиках хранятся в виде документов, таблиц и бумаг, которые невозможно обновить и проверить в режиме реального времени, что не способствует длительному хранению, эффективность работы супермаркета очень низкая.

Система управления продуктовым магазином начинается с реальной ситуации работы и управления предприятием, сочетает в себе текущие тенденции развития розничной торговли Китая, соответствует требованиям розничной торговли к информатизации и обеспечивает систематическое и комплексное техническое решение. По вышеуказанным причинам системы управления супермаркетами играют большую роль в различных коммерческих сферах и применяются во все большем количестве крупных и средних супермаркетов.

Небольшие супермаркеты и продуктовые магазины не смогли увидеть преимущества, которые дает система управления за короткий период времени, и могут даже почувствовать, что это ненужно и даже более хлопотно, ограничено по финансовым и другим причинам из-за использования дорогостоящих систем управления супермаркетами.

Система управления продуктовым магазином может автоматизировать и стандартизировать управление продуктовым магазином, тем самым реализуя полностью автоматизированное управление внутри продуктового магазина, эффективно используя различные ресурсы, удаляя избыточные функции и делая систему более удобной и простой в использовании. Эффективность и лучшее обслуживание людей – это проблемы, которые срочно должна решить нынешняя система управления продуктами магазинами.

1 ОБЩАЯ ХАРАКТЕРИСТИКА ИССЛЕДУЕМОЙ ЗАДАЧИ

1.1 Предметная область и объект проводимого исследования

С 21-го века информационные технологии быстро развиваются. ИТ-системы, представленные системами управления информацией, быстро вошли во все сферы жизни, влияя на методы работы традиционных отраслей и повышая эффективность производства. Среди них продуктовые магазины неизбежно используют компьютеры. Ежедневный поток людей в продуктовых магазинах увеличился.

В продуктовых магазинах много видов товаров и сложное управление. В результате многие товары находятся на грани истечения срока годности и просрочены. Кроме того, предстоящие и просроченные товары снизят энтузиазм потребителей к покупке. Одних только бухгалтерских книг уже недостаточно. Ведется ежедневный учет, а бухгалтерские книги громоздки в использовании, их трудно проверить и исправить в случае возникновения ошибок. Со временем образуется порочный круг. Системы управления на рынке являются дорогостоящими, и небольшие продуктовые магазины просто не могут себе позволить такие высокие закупочные цены. В этом заключается основная проблема, которую необходимо решить.

Исходя из этого, предлагается разработать систему, которая вводит все данные в базу данных, а затем использует систему для управления этими данными и получает различную полезную информацию посредством статистики и анализа, так что менеджерам продуктовых магазинов не нужно приобрести дорогостоящее оборудование для управления. Программное обеспечение также может использовать аналогичные функции.

Основная цель системы управления продуктовым магазином повысить эффективность управления супермаркетами и облегчить менеджерам управление продуктами, запасами и данными о продажах. Проблемы, которые должна решить система управления продуктовым магазином, в основном сосредоточены на реагировании на запросы системных функций, мониторинге запасов

в реальном времени и оптимизации процесса оформления заказа.

Бизнес-процесс продуктового магазина обычно включает в себя множество звеньев, таких как закупки, управление запасами, продажи и возвраты. Каждое звено требует эффективного управления информацией для повышения производительности, сокращения ошибок и обеспечения точности и актуальности данных.

С развитием экономики на городских и сельских улицах и в общинах появилось множество супермаркетов. Хотя малые и средние супермаркеты становятся все более распространенными, в их методах управления по-прежнему преобладает ручное управление. Большое количество заказов на товары и генерируемая информация о поставщиках хранятся в виде документов, таблиц и бумаги, которые невозможно обновляется и проверяется в режиме реального времени, что не способствует длительному хранению, эффективность работы супермаркета очень низкая. Это проблема и продуктовых магазинов.

Система управления продуктовым магазином начинается с реальной ситуации работы и управления предприятием, сочетает в себе текущие тенденции развития розничной торговли Китая, соответствует требованиям розничной торговли к информатизации и обеспечивает систематическое и комплексное техническое решение. По вышеуказанным причинам системы управления супермаркетами играют большую роль в различных коммерческих сферах и применяются во все большем количестве крупных и средних супермаркетов. Однако, согласно опросам, хотя системы управления супермаркетами широко используется в крупных и средних супермаркетах. Он широко используется и ценится, но редко используется в небольших супермаркетах и продуктовых магазинах и не получил большого внимания.

Небольшие супермаркеты и продуктовые магазины не смогли увидеть преимущества, которые дает система управления за короткий период времени, и могут даже почувствовать, что это ненужно и даже более хлопотно. ограничено по финансовым и другим причинам из-за использования дорогостоящих систем управления супермаркетами.

Система управления продуктовым магазином может автоматизировать и стандартизировать управление продуктовым магазином, тем самым реализуя полностью автоматизированное управление внутри продуктового магазина, эффективно используя различные ресурсы, удаляя избыточные функции и делая систему более удобной и простой в использовании. Эффективность и лучшее обслуживание людей – это проблемы, которые срочно должна решить нынешняя система управления продуктовыми магазинами.

Система управления продуктовым магазином должна охватывать множество функциональных модулей, таких как управление продукцией, управление клиентами, статистика продаж, оповещение о запасах и т. д. Эти функции должны быть бесшовно интегрированы для удовлетворения потребностей управления в различных сценариях.

Система должна иметь полноценный механизм защиты данных для предотвращения утечки или фальсификации данных. Кроме того, необходимо также рассмотреть вопрос защиты конфиденциальности клиентов, чтобы гарантировать, что информация о клиентах не будет использоваться не по назначению.

Повышение эффективности управления продуктовым магазином:

- реализовать автоматизацию управления продуктами, включая такие операции, как добавление, изменение, удаление и запрос продуктов;
- мониторинг товарных запасов в режиме реального времени, включая складирование, исходящую доставку, распределение и другие операции;
- запись и анализ продажи продукции, а также создание отчетов о продажах для облегчения анализа принятия решений менеджерами.

Улучшить уровень обслуживания:

- поддерживает несколько способов оплаты, таких как наличные, кредитная карта, мобильный платеж и т.д., обеспечивая удобство оплаты;
- на основе анализа данных о продажах прогноз будущих тенденций продаж, чтобы помочь продуктовым магазинам принимать более точные решения о покупке.

Повышение удовлетворенности клиентов:

- с помощью функции управления членством предоставление участникам персонализированные услуги, такие как накопление баллов и скидки, для повышения лояльности клиентов;
- автоматическое создание различных финансовых отчетов, таких как ежедневные отчеты о продажах, ежемесячные отчеты о прибыли и т.д., для упрощения финансовых процессов;
- предоставление подробных отчетов о продажах, которые помогут сформулировать маркетинговую стратегию и оптимизировать операционные процессы.

Гарантия безопасности данных:

- обеспечить безопасность данных и принять необходимые меры безопасности для предотвращения утечки и повреждения данных;
- система должна иметь хорошую масштабируемость, чтобы адаптироваться к будущим изменениям потребностей бизнеса;
- интерфейс прост, понятен и удобен в использовании, что обеспечивает удобство использования системы.

Оптимизация финансового управления:

- автоматически генерировать различные финансовые отчеты для облегчения проверки счетов и финансового анализа финансовым персоналом;
- предоставление точных данных о продажах для лучшего контроля затрат и анализа прибыли;
- поддержание нескольких методов расчета для снижения финансовых рисков от источника.

Задачи, которые должна решить система управления продуктовым магазином:

Ответ о функциональных требованиях системы:

- система должна быть адаптирована и разработана на основе реальных оперативных потребностей для удовлетворения конкретных потребностей различных магазинов повседневного спроса;

- необходимо объединить внешнюю техническую архитектуру и внутреннюю структуру базы данных, чтобы обеспечить общую координацию и производительность системы.

Мониторинг запасов в режиме реального времени:

- отслеживание запасов продукции в режиме реального времени, вовремя пополнение или сокращение запасов и избегание нехватки или излишка продукции;

- автоматический вход и выход товаров и оперативное напоминание о недостаточности или избытке запасов.

Оптимизация процесса оформления заказа:

- сокращение времени ожидания клиентов, повышение эффективности касс и повышение удовлетворенности клиентов;

- поддержание несколько способов оплаты для обеспечения быстрого завершения транзакций.

Анализ данных о продажах:

- предоставление ежедневных, еженедельных или ежемесячных отчетов о продажах, чтобы помочь менеджерам понять тенденции продаж;

- анализ данные о продажах, чтобы сформулировать маркетинговую стратегию и оптимизировать операционные процессы.

Защита данных:

- принятие необходимых мер безопасности для предотвращения утечки и повреждения данных;

- регулярно выполнять резервное копирование и обслуживание системы для обеспечения целостности и безопасности данных.

Улучшение масштабируемости системы:

- система должна иметь хорошую масштабируемость, чтобы адаптироваться к возможным изменениям потребностей бизнеса в будущем;

- при проектировании необходимо учитывать добавление будущих функций, чтобы обеспечить возможность гибкой настройки и обновления системы.

Организационная структура продуктового магазина обычно определяется функциональностью и эффективностью. Вот простой пример организационной структуры и описание основных обязанностей каждой роли:

- владелец/менеджер магазина: отвечает за общую работу и управление магазином, чтобы обеспечить бесперебойную ежедневную работу магазина. Помимо прочего, формулирование бизнес-стратегии магазина, целей продаж и планов реализации;

- финансы: управление финансовым состоянием магазина, включая составление бюджета, контроль затрат, учет доходов и расходов и т.д. Необходимо убедиться, что магазин финансово здоров и может продолжать свою деятельность;

- закупки: на основе данных о продажах и рыночных тенденциях нужно разработать планы закупок, выбрать подходящих поставщиков товаров и управлять уровнями запасов, чтобы избежать чрезмерных запасов или их нехватки;

- демонстрация продуктов: нужно убедиться, что все продукты правильно размещены на полках, имеют четкие цены и правильно выставлены, чтобы привлечь клиентов к покупке;

- руководство по покупкам: консультирование клиентов, помощь клиентам в выборе нужных им продуктов, обработка жалоб клиентов и обеспечение удовлетворенности клиентов;

- кассир: отвечает за работу кассира, включая расчет продуктов, сбор платежей и операции по обмену, для обеспечения точности и эффективности транзакций;

- обслуживающий персонал: регулярно должен организовывать и поддерживать инвентарь товаров для обеспечения свежести и качества товаров и снижения потерь;

- маркетинговый персонал: разрабатывает и реализует стратегии продвижения магазина, такие как скидки, подарки и т.д., для увеличения возможностей продаж.

Срок разработки системы управления продуктовым магазином составляет около 40 дней с учетом того, что среднемесячная зарплата инженера-программиста составляет 70000 рублей, а плата за Интернет на 40 дней – 10 000 рублей. Эта система использует инструменты разработки IDEA и базу данных MySQL. Среда SpringBoot на данный момент популярна, бесплатна и имеет открытый исходный код. Поэтому в процессе разработки стоимость проекта составляет около 70000 рублей, а зарплата сотрудников продуктового магазина – около 40000 рублей. Рассчитана ежемесячная чистая прибыль продуктового магазина около 50000 руб. Таким образом, предполагается, что средства, использованные на разработку, можно будет вернуть примерно через 2-3 месяца после ввода системы в эксплуатацию.

Система управления продуктовым магазином проста в эксплуатации и удобна в использовании. Систему можно использовать на обычных домашних компьютерах, причем систему можно использовать в обычном режиме без подключения к Интернету. Менеджерам не нужно много знать о компьютерах. Им нужно только знать основы работы с компьютером и просто нажимать в соответствии с функциями системы.

Что касается аппаратного обеспечения, типичный домашний компьютер может поддерживать разработку и последующее использование системы. Программное обеспечение использует IDEA, а база данных – MySQL. Платформа SpringBoot проста и удобна в использовании, и разработчики, которые во время учебы много раз использовали эти два программного обеспечения для подключения баз данных и проектов, могут использовать ее без каких-либо препятствий.

Локальные нормативные акты, регулирующие организационную структуру образовательных учреждений, обеспечивают основу для решения организационных вопросов, связанных с существующими нормативными процессами. Указывая структурные подразделения, эти положения усиливают распределение ролей и функций между различными отделами внутри агентства.

Положение об управлении продуктовыми магазинами и порядок их работы «Закон о защите потребителей» основано на общих положениях коммерческого законодательства «Закон о безопасности пищевых продуктов» и определяет состав, цели, задачи и функции контролирующего персонала.

Основными задачами надзорных органов являются:

Контролировать качество и безопасность товаров продуктового магазина на предмет соблюдения гигиенических норм.

В состав контролирующего персонала могут входить:

- бюро промышленности и торговли;
- налоговая инспекция;
- санитарно-эпидемические станции;
- табачное бюро;
- бюро городского управления и административного правопорядка;
- агентство по охране окружающей среды;
- бюро городского планирования;
- бюро общественной безопасности.

Список является гибким и его состав будет меняться в ходе разработки модуля из-за привлечения неоправданно большого количества ответственных лиц.

Далее обсудим правила внутреннего распорядка продуктового магазина, регулирующие порядок приобретения, хранения и реализации товаров:

- товары в продуктовом магазине должны соответствовать установленным гигиеническим нормам;
- оформление покупки продуктов питания осуществляется покупателями на основании ежедневных продаж;
- управляющий магазином/управляющий магазином обеспечивает нормальную повседневную работу продуктового магазина;
- внутренний отдел контроля качества гарантирует, что в продаже не будет просроченной или просроченной продукции;
- гиды по магазинам должны относиться к клиентам с энтузиазмом

и терпением;

Система управления продуктовым магазином представляет собой комплексное решение для автоматизации подсчета, хранения, обработки и анализа информации о работе продуктового магазина. Включая статистику продаж, статистику покупок, статистику возвратов и другие важные функции процесса.

Информационная система также включает в себя личный кабинет для кассиров и управляющих/менеджеров магазинов. В этой системе менеджер магазина может проверять ежедневную деловую ситуацию и производить соответствующую обработку, а кассир может выполнять кассовые операции в этой системе.

Кроме того, обращайте внимание на рабочие характеристики во время реальных операций, чтобы обеспечить точность каждого этапа операции и предотвратить жалобы клиентов из-за неправильной эксплуатации.

1.2 Обзор существующих методов решения аналогичных задач

При создании комплексной и эффективной системы управления продуктовым магазином команде разработчиков необходимо тщательно продумать и решить ряд сложных вопросов, чтобы гарантировать, что система сможет не только беспрепятственно интегрироваться в существующие бизнес-процессы, но и значительно повысить эффективность работы и удовлетворенность пользователей. Ниже приводится подробное и расширенное объяснение ключевых соображений:

– функции, специфичные для должностей, и управление полномочиями: При проектировании системы необходимо четко разделить функции и полномочия различных должностей (например, кассиров, операционистов, управляющих магазинами и менеджеров по запасам и т. д.), чтобы гарантировать, что каждый сотрудник может получить доступ только к тем данным и функциям, которые напрямую связаны с его обязанностями. Например, кассиры должны сосредоточиться на быстром оформлении заказов и обработке плате-

жей, в то время как операционисты должны управлять финансовыми потоками и составлять отчеты, а менеджеры магазинов должны всесторонне отслеживать показатели продаж, состояние запасов и эффективность работы сотрудников. Внедряя управление доступом на основе ролей (RBAC), мы можем защитить безопасность конфиденциальной информации и повысить эффективность работы;

- оптимизация процесса совершения покупок клиентами: для повышения удовлетворенности клиентов система должна интегрировать удобные функции самообслуживания, такие как каналы быстрого оформления заказа, запрос баллов участника, использование электронных купонов и т. д., а также поддерживать различные способы оплаты, включая наличные, кредитные карты, мобильные платежи (например, WeChat Pay, Alipay), членские карты и т. д., чтобы сократить время ожидания в очередях и сделать процесс совершения покупок более плавным и приятным. Кроме того, для улучшения покупательского опыта и повышения лояльности клиентов предлагаются персонализированные рекомендации и акции;

- комплексные и мощные системные функции: основные функции системы должны охватывать основные операции, такие как сканирование продуктов, автоматический расчет цен, диверсифицированная обработка платежей и мгновенная печать чеков. В то же время она должна глубоко интегрировать управление запасами (обновление запасов в реальном времени, предупреждения о низком уровне запасов), управление жизненным циклом продукта (напоминания об истечении срока годности, отслеживание партий), управление отношениями с поставщиками (отслеживание заказов, управление контрактами) и комплексные функции финансового управления (записи доходов и расходов, анализ прибыли, налоговые отчеты). Интеграция этих функций может не только повысить эффективность повседневной деятельности, но и обеспечить информационную поддержку для принятия решений и способствовать росту бизнеса;

– совместимость оборудования и интеграция программного обеспечения: система должна широко поддерживать различные аппаратные устройства POS (например, POS-терминалы с сенсорным экраном, сканеры штрих-кодов) и легко подключаться к стороннему бизнес-программному обеспечению (например, бухгалтерскому программному обеспечению, системам ERP, платформам электронной коммерции);

– безопасность данных и механизм резервного копирования: поскольку данные являются жизненно важными для предприятия, система должна использовать передовую технологию шифрования данных (такую как протокол SSL/TLS) для защиты всей конфиденциальной информации (особенно информации о платежах клиентов), а также выполнять регулярное резервное копирование в облаке или на физическом сервере для предотвращения потери или повреждения данных;

– интуитивно понятный и простой в использовании пользовательский интерфейс: пользовательский интерфейс разработан таким образом, чтобы быть простым и интуитивно понятным, с использованием графических руководств по эксплуатации и модальных диалоговых окон, помогающих пользователям выполнять различные задачи, что снижает затраты на обучение и количество ошибок при эксплуатации. Благодаря обратной связи с пользователями мы постоянно оптимизируем дизайн UI/UX, чтобы даже новички могли быстро приступить к работе и повысить эффективность работы;

– высокая производительность и надежность: с точки зрения производительности система должна использовать эффективные алгоритмы и оптимизированные структуры баз данных, чтобы гарантировать быстрое время отклика даже в часы пик (например, во время праздничных акций) и достигать скорости обработки в считанные секунды. В то же время создан комплексный механизм мониторинга и устранения неисправностей, позволяющий минимизировать время простоя системы и обеспечить непрерывность бизнеса за счет избыточной конструкции и технологии автоматического переключения при сбоях;

– гибкая масштабируемость функций: по мере развития бизнеса система должна иметь возможность легко добавлять новые функциональные модули (например, интеграцию с онлайн-торговыми центрами, систему управления взаимоотношениями с клиентами (CRM)) или корректировать существующие функции в соответствии с изменениями рынка, чтобы система оставалась перспективной и конкурентоспособной. Достичь этой цели можно с помощью внедрения архитектуры микросервисов или принципов модульного проектирования;

– соблюдение нормативных требований и правовых норм: убедитесь, что проектирование и реализация системы строго соответствуют местным законам и правилам, включая, помимо прочего, законы о защите данных (GDPR, CCPA и т. д.), законы о защите прав потребителей, стандарты безопасности электронных платежей (PCI DSS) и т. д., чтобы избежать правовых рисков и завоевать доверие клиентов.

В современной розничной торговле качество систем управления продуктовыми магазинами напрямую влияет на эффективность работы, устойчивость бизнес-процессов и уровень удовлетворенности клиентов компании. Использование современных ИТ-решений позволяет оптимизировать управление запасами, автоматизировать продажи, упрощать логистику и повышать точность аналитики. Существует три популярные системы управления, широко применяемые в отрасли: SAP, Oracle Retail и Microsoft Dynamics 365 Commerce. Ниже приведены их ключевые преимущества и недостатки, отражённые на рисунке 1.



Рисунок 1 – Система управления SAP

SAP – всемирно известный поставщик программного обеспечения для управления предприятием. Его система управления продуктовым магазином охватывает различные бизнес-связь, такие как закупки, продажи, инвентарь и финансы, предоставляя комплексное решение для управления продуктовыми магазинами. Система известна своей высокой степенью интеграции и масштабируемостью и может адаптироваться к потребностям продуктовых магазинов разных размеров и бизнес-моделей.

Анализ преимуществ:

- высокая степень интеграции: система SAP может органично интегрировать различные бизнес-модули продуктового магазина, реализовать обмен данными и взаимодействие в режиме реального времени, избежать создания информационных островов и повысить общую эффективность работы;

- богатый набор функциональных модулей: система предоставляет широкий спектр функциональных модулей, таких как управление цепочками поставок, управление взаимоотношениями с клиентами, финансовое управление и т. д., которые могут удовлетворить потребности в управлении продуктовыми магазинами в различных сферах бизнеса;

- мощные возможности анализа данных: система SAP обладает передовыми инструментами анализа данных, которые позволяют проводить глубокий анализ огромных объемов данных, обеспечивая надежную поддержку для принятия корпоративных решений. например, анализируя данные о продажах, компании могут понять тенденции продаж и потребности клиентов в различных товарах, а также скорректировать свои стратегии закупок и маркетинга.

Ограничения:

- высокая стоимость внедрения: Из-за сложности и мощных функций системы SAP процесс ее внедрения требует больших трудозатрат, материальных и финансовых ресурсов, включая затраты на закупку программного обеспечения, модернизацию оборудования, обучение персонала и т. д. Это может

оказаться существенным бременем для некоторых небольших продуктовых магазинов;

– сложность настройки системы: хотя система SAP обладает высокой масштабируемостью, в реальных условиях ее настройка в соответствии с конкретными потребностями предприятия может столкнуться с определенными техническими трудностями и потребовать привлечения профессиональных специалистов и длительного периода времени.

Технология Oracle Retail изображена на рисунке 2.



Рисунок 2 – Система управления Oracle Retail

Oracle Retail – это комплексная система управления, разработанная Oracle для розничной торговли, которая ориентирована на предоставление эффективных решений по управлению операциями для предприятий розничной торговли. Система имеет широкий спектр применения в секторе розничной торговли, особенно при управлении крупными сетями продуктовых магазинов.

Анализ преимуществ:

– сильная отраслевая специализация: система Oracle Retail разработана на основе характеристик и потребностей розничной торговли. Она имеет глубокое понимание бизнес-модели и процессов управления продуктовыми магазинами и может предоставлять более практичные решения. Например, система разработала специальный модуль управления запасами, который учитывает широкий ассортимент товаров и позволяет решать сложные задачи

управления запасами в продуктовых магазинах, что позволяет добиться более совершенного управления товарами;

- гибкие методы развертывания: система поддерживает несколько методов развертывания, включая локальное развертывание и облачное развертывание. Предприятия могут выбрать подходящий метод развертывания в зависимости от их реальной ситуации. Облачное развертывание имеет такие преимущества, как низкая стоимость и простота обслуживания, и подходит для малых и средних предприятий; локальное развертывание может лучше соответствовать требованиям компании к безопасности и конфиденциальности данных и подходит для крупных предприятий с высокими требованиями к безопасности данных;

- хороший пользовательский опыт: система Oracle Retail ориентирована на пользовательский опыт, имеет простой и интуитивно понятный интерфейс, а также удобную и быструю работу. Сотрудники могут за короткий промежуток времени освоить систему и научиться ею пользоваться, тем самым повышая эффективность работы.

Ограничения:

- обновления системы несвоевременны: в некоторых случаях обновления системы Oracle Retail могут не поспевать за изменениями рынка и технологическими разработками, в результате чего некоторые функции не могут удовлетворить новые потребности предприятия. Это может повлиять на конкурентоспособность и эффективность работы бизнеса;

- высокие требования к технической поддержке: в связи со сложными функциями системы в процессе ее использования могут возникать различные технические проблемы. Поэтому предприятиям необходимо иметь определенную техническую мощь и профессиональных технических специалистов для поддержания и поддержки нормальной работы системы, что может стать проблемой для некоторых предприятий со слабыми техническими возможностями.

Microsoft Dynamics 365 Commerce – это облачная система управления

розничной торговлей, запущенная корпорацией Microsoft, изображена на рисунке 3.



Рисунок 3 – Система управления Microsoft Dynamics 365 Commerce

Она объединяет традиционные функции управления розничной торговлей с современными цифровыми технологиями для предоставления инновационных решений по управлению для продуктовых магазинов. Система отличается гибкостью и масштабируемостью и может удовлетворить потребности продуктовых магазинов разных размеров и стадий развития.

Анализ преимуществ:

- облачная архитектура: Microsoft Dynamics 365 Commerce использует облачную архитектуру, которая отличается высокой гибкостью и масштабируемостью. Предприятия могут в любое время настраивать функции системы и конфигурации ресурсов в соответствии с потребностями своего бизнеса, не беспокоясь об обновлении и обслуживании аппаратного обеспечения. В то же время облачные сервисы также могут обеспечить более высокую безопасность и надежность данных, гарантируя безопасность информационных активов компании.

- полная интеграция с экосистемой Microsoft: как продукт Microsoft, система может быть легко интегрирована с другим программным обеспечением и службами Microsoft Office (такими как Office 365, Azure и т. д.) для обеспечения обмена данными и совместной работы. Например, сотрудники могут об-

рабатывать файлы и данные, связанные с деятельностью продуктового магазина, непосредственно в привычном интерфейсе Office, что повышает эффективность работы.

– инновационные технологические приложения: Microsoft Dynamics 365 Commerce активно внедряет новые технологии, такие как искусственный интеллект и большие данные, принося больше инноваций и возможностей в управление операциями продуктовых магазинов. Например, технологию искусственного интеллекта можно использовать для реализации таких функций, как интеллектуальное обслуживание клиентов и интеллектуальные рекомендации, тем самым улучшая опыт покупок для клиентов и эффективность продаж компании.

Ограничения:

– высокая зависимость от Интернета: поскольку система основана на облачных вычислениях, для обеспечения нормальной работы системы требуется стабильное сетевое соединение. В случае ненадлежащего качества сетевой среды могут возникнуть такие проблемы, как зависание системы и потеря данных, что повлияет на нормальную работу предприятия.

– проблемы конфиденциальности данных: хотя Microsoft приняла ряд мер для защиты конфиденциальности и безопасности пользовательских данных, в среде облачных вычислений данные хранятся на облачных серверах, и по-прежнему существуют определенные риски конфиденциальности данных. Некоторым компаниям, предъявляющим чрезвычайно высокие требования к конфиденциальности данных, может потребоваться дополнительно рассмотреть вопросы безопасности данных.

Подводя итог, можно сказать, что качество системы управления продуктовым магазином играет жизненно важную роль в современной розничной торговле, поскольку напрямую связано с эффективностью работы компании и удовлетворенностью клиентов. SAP, Oracle Retail и Microsoft Dynamics 365 Commerce – три популярные системы управления, каждая из которых имеет

свои уникальные преимущества и ограничения. При выборе подходящей системы управления предприятия должны в полной мере учитывать такие факторы, как масштаб предприятия, потребности бизнеса, технический уровень и бюджет, взвешивать преимущества и недостатки различных систем и выбирать наиболее подходящий им инструмент управления.

Интерфейс системы прост и удобен в использовании. Даже без профессиональной подготовки вы сможете быстро приступить к работе и добиться более эффективного управления. По сравнению с другими сложными системами она имеет более высокие преимущества и предоставляет малым и средним предприятиям предпочтительные и качественные решения.

В то же время, с постоянным развитием технологий и изменениями на рынке, системы управления продуктовыми магазинами также должны продолжать внедрять инновации и совершенствоваться, чтобы адаптироваться к все более сложной розничной среде и оказывать более мощную поддержку развитию предприятия. Дальнейшие исследования могут быть сосредоточены на том, как оптимизировать функции систем управления продуктовыми магазинами, повысить их производительность и безопасность, а также как лучше интегрировать новые технологии для создания большей ценности для предприятий розничной торговли.

2 АЛГОРИТМИЧЕСКИЕ И ПРОГРАММНЫЕ РЕШЕНИЯ ПО УПРАВЛЕНИЮ МАГАЗИНОМ

2.1 Модульная структура приложения и его функциональные зависимости

В процессе проектирования и внедрения системы управления продуктовым магазином на основе фреймворка Spring Boot необходимо сосредоточиться на интеграции моделирования бизнес-логики и механизмов оптимизации интеллектуальных алгоритмов, а также добиться функциональной связки и повышения производительности за счет модульной архитектуры. Основные функциональные модули системы и соответствующие им алгоритмы/технические решения, описанные ниже.

Модуль управления закупками для небольших продуктовых магазинов необходимо разрабатывать с учетом основных целей: «легкость, интуитивность и простота в эксплуатации», а также реализовать автоматизированную поддержку принятия решений на основе базовых данных. Система начинает работу с мониторинга запасов в режиме реального времени и устанавливает динамический страховой запас для каждой категории товаров. Например, система устанавливает динамический страховой запас на основе среднего объема продаж за последние 7 дней, умноженного на коэффициент эластичности 1,2 от максимального количества дней задержки поставщика (например, 2 дня). Когда уровень запасов падает ниже порогового значения, автоматически срабатывает желтое предупреждение, а товары, которые необходимо срочно пополнить, выделяются в верхней части интерфейса. Количество пополнения рассчитывается с использованием алгоритма прогрессивного средневзвешенного значения, который дает 50 % веса объему продаж за последние три дня и 30 % веса историческим данным за тот же период. Он объединяется с прогнозом продаж на следующие три дня для генерации рекомендуемого количества покупки, что позволяет избежать задержки простого усреднения и может адаптироваться к характеристикам колебаний продаж небольших магазинов.

Модуль выбора поставщика имеет встроенную панель сравнения цен, которая автоматически фиксирует последние три предложения на аналогичные продукты от каждого поставщика для создания кривой ценового тренда. Для чувствительных к сроку годности продуктов система автоматически вычисляет теоретическую дату истечения срока годности при сканировании кода для поступления на склад, отмечает ее оранжевым цветом в списке инвентаря за 30 дней и разумно сокращает количество пополнения аналогичных продуктов в рекомендациях по закупкам, чтобы избежать задержек.

После определения категории пополнения система формирует рекомендуемое количество закупки на основе заданной формулы пополнения (последний объем продаж $\times$ цикл пополнения $\times$ коэффициент эластичности 1,2), помогая владельцам магазинов постоянно оптимизировать свои бизнес-стратегии, Схема рабочего процесса представлена на рисунке 4.

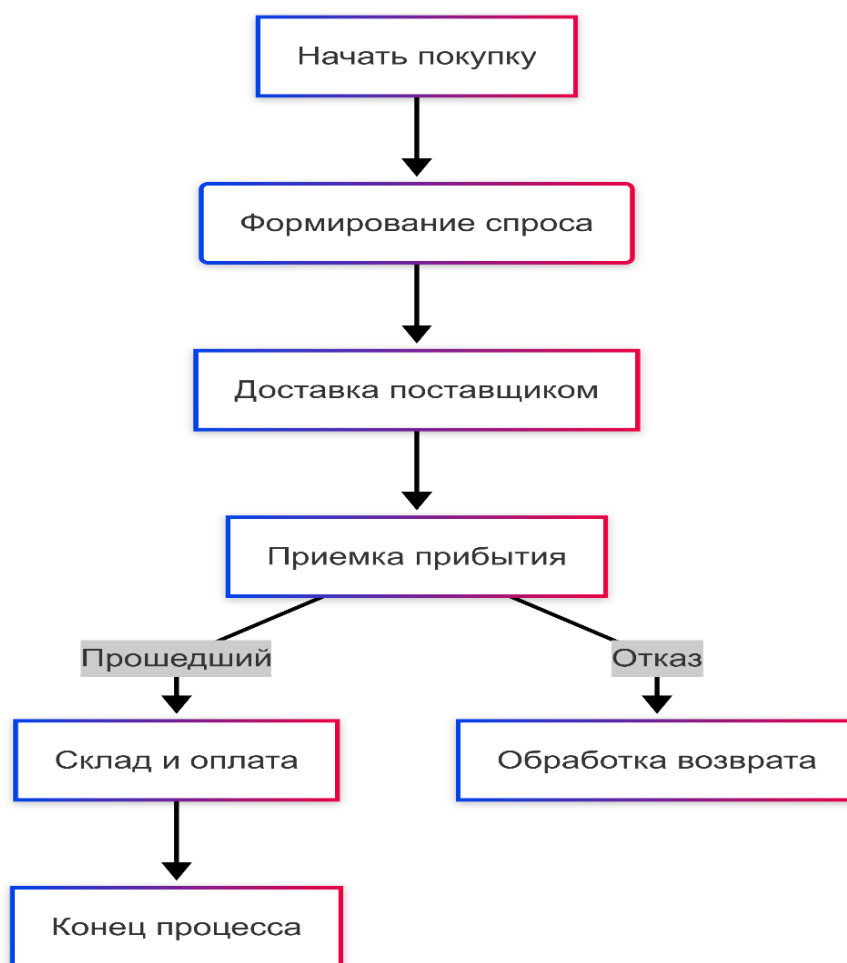


Рисунок 4 – Диаграмма рабочего процесса управления закупками

Модуль управления продажами. Основной рабочий процесс модуля продаж небольшого продуктового магазина вращается вокруг замкнутого цикла «транзакции в реальном времени – обратная связь по данным – поддержка принятия решений». Весь процесс не требует сложного переключения интерфейсов и полностью интегрирован в основной интерфейс кассира. В момент сканирования штрих-кода продукта система автоматически запускает многомерную обработку: сначала вызывается прайс-лист в базе данных для сопоставления текущей цены продукта. Если обнаруживается, что клиент ранее зарегистрировал номер мобильного телефона (например, последние четыре введенные цифры совпадают с информацией о членстве), скидка члена (например, 9,5 %) немедленно добавляется, и отображается сравнение между исходной ценой и ценой со скидкой.

Логика продвижения встроена в процесс транзакции и автоматически сопоставляется с помощью механизма правил. Алгоритм продвижения использует механизм приоритетного покрытия: когда несколько видов деятельности пересекаются, система автоматически выбирает план с наибольшей выгодой для клиентов. Например, когда одновременно выполняются оба условия: «3 скидки на заказы свыше 30 юаней» и «8 скидок на заказы свыше 50 юаней», последнее будет выполнено первым, поскольку предполагает большую скидку. При этом владелец магазина получит уведомление с аналитической подсказкой: на основе анализа исторических данных система сообщает, что «скидка на этот заказ на 15 % выше, чем среднесуточная».

Если условия нескольких акций равнозначны по выгоде, система дополнительно учитывает предпочтения клиентов, историю покупок, а также частоту использования каждой из акций, чтобы предложить наиболее релевантную. Таким образом, алгоритм не только обеспечивает максимальную ценность для клиента, но и способствует повышению конверсии и лояльности. Все действия системы логируются и могут быть просмотрены в панели администратора для последующего анализа и корректировки маркетинговых стратегий.

После накопления данных о продажах система автоматически выполняет аналитические задачи после закрытия каждого дня. Основной алгоритм включает два измерения: одно из них – генерация кривой тренда продаж каждой категории с помощью метода скользящей средней за 7 дней и автоматическая отметка аномальных значений с колебаниями, превышающими $\pm 30\%$; другое – реализация метода классификации ABC и разделение товаров на класс А (верхние 70 %), класс В (средние 25 %) и класс С (нижние 5 %) в соответствии с их вкладом в продажи. Товары класса С автоматически добавляются в список мониторинга медленно реализуемых товаров. Нажатие на любой аномальный индикатор (например, «продажи лапши быстрого приготовления упали на 40 % в годовом исчислении») может напрямую перейти к интерфейсу предложения по пополнению запасов модуля закупок, образуя замкнутый цикл принятия решений от продаж до закупок. Диаграмма рабочего процесса показана на рисунке 5.

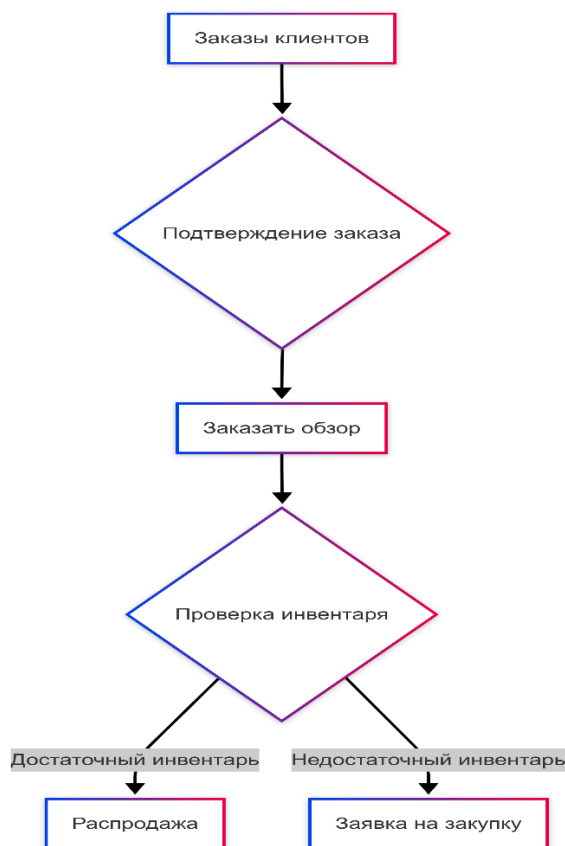


Рисунок 5 – Схема рабочего процесса модуля управления продажами

Модуль управления заказами. Когда клиент завершает оформление заказа, сканируя код, система автоматически генерирует структурированные данные заказа на продажу – каждый заказ содержит четыре основных поля: временную метку (с точностью до секунды), список товаров (включая цену за единицу и количество), способ оплаты (наличные/ WeChat/ Alipay) и номер мобильного телефона участника (если есть), а также генерирует уникальный код заказа с помощью алгоритма хеширования (например, 20230815163245_3589, где первые 14 цифр – время, а последние 4 цифры случайные числа). Эти данные хранятся в режиме реального времени в локальной зашифрованной базе данных SQLite мобильного телефона и хранятся в ежедневных разделах, что гарантирует сохранность записей транзакций даже в случае сбоя сети.

Обработка данных заказов использует легкий режим потоковых вычислений: каждый раз, когда завершаются 5 транзакций, система автоматически запускает инкрементальный анализ. Эти показатели отображаются в виде визуальной пузырьковой диаграммы на боковой панели интерфейса кассового аппарата. Размер пузырька представляет объем продаж, а глубина цвета отражает маржу прибыли.

Если требуется возврат или обмен, необходимо отсканировать QR-код исходного заказа (или введите временной диапазон для поиска), система извлекает данные исходного заказа и автоматически рассчитывает максимальную сумму возврата (не превышающую исходную фактическую полученную сумму). Все записи о возвратах хранятся независимо.

В случае возврата система дополнительно проверяет, были ли использованы скидки, бонусы или промокоды, и при необходимости пересчитывает итоговую сумму с учетом применённых условий. Средства возвращаются тем же способом, которым была произведена оплата, за исключением случаев, когда пользователь выбирает альтернативный метод. Все действия, связанные с возвратами и обменами, протоколируются и доступны для просмотра в раз-

деле «История операций». Кроме того, в системе предусмотрена автоматическая блокировка возврата, если заказ уже участвует в процессе внутреннего аудита или был помечен как подозрительный, о чём кассир получает уведомление в реальном времени.

Все данные о заказах поддерживают нечеткий поиск. Введя фрагмент названия продукта и диапазон дат, вы можете быстро найти связанные заказы и экспортировать их в упрощенную электронную книгу, которая соответствует налоговым требованиям. Схема рабочего процесса показана на рисунке 6.

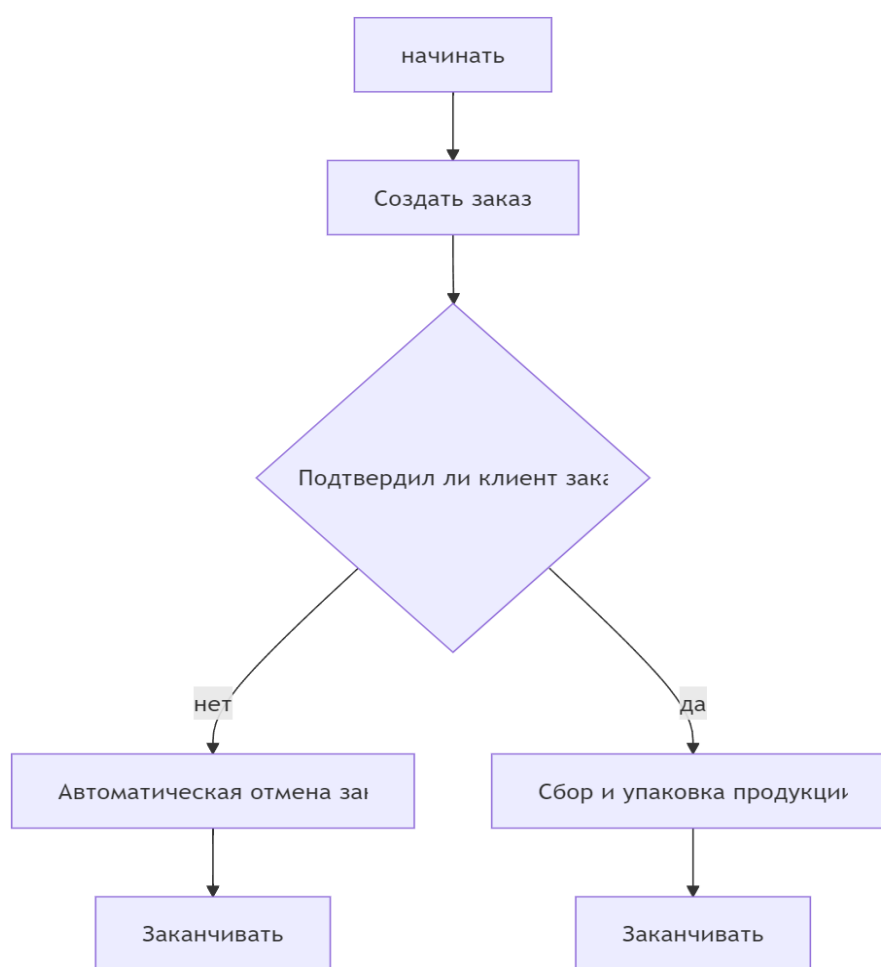


Рисунок 6 – Схема рабочего процесса модуля управления заказами

Модуль управления продуктом. При открытии интерфейса управления продуктами на главной странице будет отображаться обновляемая в реальном времени панель основных показателей: верхнее поле поиска поддерживает нечеткое соответствие названий продуктов. После нажатия на любую карточку

продукта базовый информационный слой отобразит жесткие параметры, такие как штрих-код продукта, текущая цена продажи и количество запасов; операционный слой данных будет интегрировать графики тенденций продаж за последние семь дней, кривые сравнения цен для аналогичных продуктов и рассчитанные дни оборачиваемости запасов; операционный слой сохранит только три физические кнопки: «Изменить цену», «Настроить запасы» и «Удалить продукт с полки». Все операции выполняются путем нажатия и скольжения, без необходимости ввода с клавиатуры.

Когда необходимо изменить цену продажи товара, система автоматически запускает механизм тройной проверки: во-первых, она сравнивает последнюю цену покупки. Если новая цена продажи ниже себестоимости, появляется красное предупреждающее окно; во-вторых, она обнаруживает колебания цен в близлежащих супермаркетах. Если цена отклоняется от средней рыночной цены более чем на 15%, отображается жёлтая строка подсказки; наконец, проверяется действительность цифрового формата, чтобы предотвратить ввод нецифровых символов.

Дополнительно система анализирует историю предыдущих изменений цен по данному товару, выявляя аномальные или слишком частые корректировки, и в случае обнаружения подозрительной активности блокирует возможность сохранения новой цены до подтверждения менеджером. При этом учитываются сезонные коэффициенты, планируемые акции и остатки на складе, чтобы минимизировать риск убытков и оптимизировать стратегию ценообразования. Все изменения цен фиксируются в журнале операций с указанием даты, времени, пользователя и причины корректировки, а также сопровождаются аналитическим комментарием о предполагаемом влиянии на общий оборот.

Все записи об изменении данных сжимаются и хранятся с использованием дифференциального алгоритма, сохраняющего три основные информации: время операции, значение до изменения и значение после изменения. Он поддерживает отслеживание до снимков данных в любое время по временной

шкале. Техническая реализация использует атомарные характеристики транзакций базы данных для обеспечения безопасности и восстанавливаемости данных. Диаграмма рабочего процесса показана на рисунке 7.

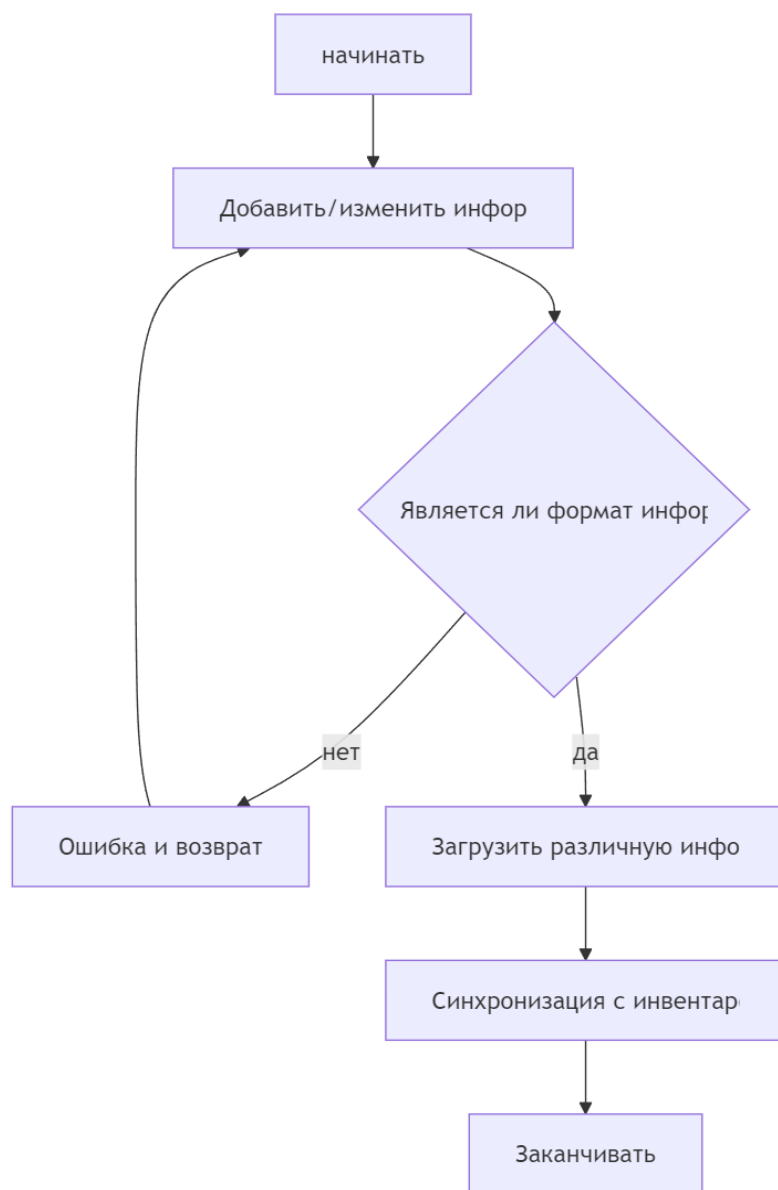


Рисунок 7 – Схема рабочего процесса модуля управления продуктом

Модуль оповещения о состоянии запасов точно фиксирует риски, связанные с запасами, с помощью мониторинга потока данных в режиме реального времени и адаптивного порогового алгоритма. При изменении запасов товаров система автоматически запускает механизм расчета динамического страхового запаса – на основе объема продаж за последние 72 часа он исключает выбросы (например, объем продаж в определенный день, превышающий

средний в 3 раза) и берет средневзвешенное значение (вес последних 24 часов составляет 60 %), умножая его на цикл пополнения (по умолчанию соответствует среднему времени прибытия поставщика) и коэффициент эластичности (начальное значение равно 1,2, которое автоматически увеличивается до 1,5, когда объем продаж колеблется более чем на 15 % в течение трех последовательных дней).

Для товаров, чувствительных к сроку годности, система одновременно рассчитывает значение риска возраста запасов – когда дата производства самой ранней партии товаров в запасах превышает $1/3$ срока годности, срабатывает оранжевая предупреждающая полоса и отображается запрос на «приоритетную продажу»; если возраст запасов превышает $1/2$ срока годности, партия запасов принудительно блокируется, и появляется предупреждающее окно с просьбой вручную подтвердить, разрешена ли продажа.

Рекомендованный рейтинг формируется на основе исторической скорости реагирования на сотрудничество. Все события предупреждения формируют журнал временной шкалы, который поддерживает скольжение для просмотра исторических записей обработки предупреждений. Элементы предупреждения, которые не обработаны вовремя, продолжают увеличивать интенсивность напоминания, от первоначального изменения цвета интерфейса до напоминания о вибрации один раз в час, пока инвентарь не вернется выше порога безопасности. Диаграмма рабочего процесса показана на рисунке 8.

Кроме того, система учитывает приоритизацию типов предупреждений: критические уведомления, связанные с жизненно важными товарными позициями, автоматически поднимаются в верхнюю часть панели и могут временно блокировать ввод новых заказов, если не устранены. Встроенный модуль обучения на основе предыдущих реакций операторов корректирует рейтинг и предлагает персонализированные рекомендации по действиям – например, временное перераспределение запасов или ускорение закупки. Все взаимодействия пользователей с системой в момент обработки предупреждения записываются, формируя основу для последующего анализа эффективности и

обновления рекомендательных алгоритмов. Система также синхронизируется с мобильными устройствами ответственных сотрудников, обеспечивая удалённый контроль и оперативное вмешательство в нештатных ситуациях.

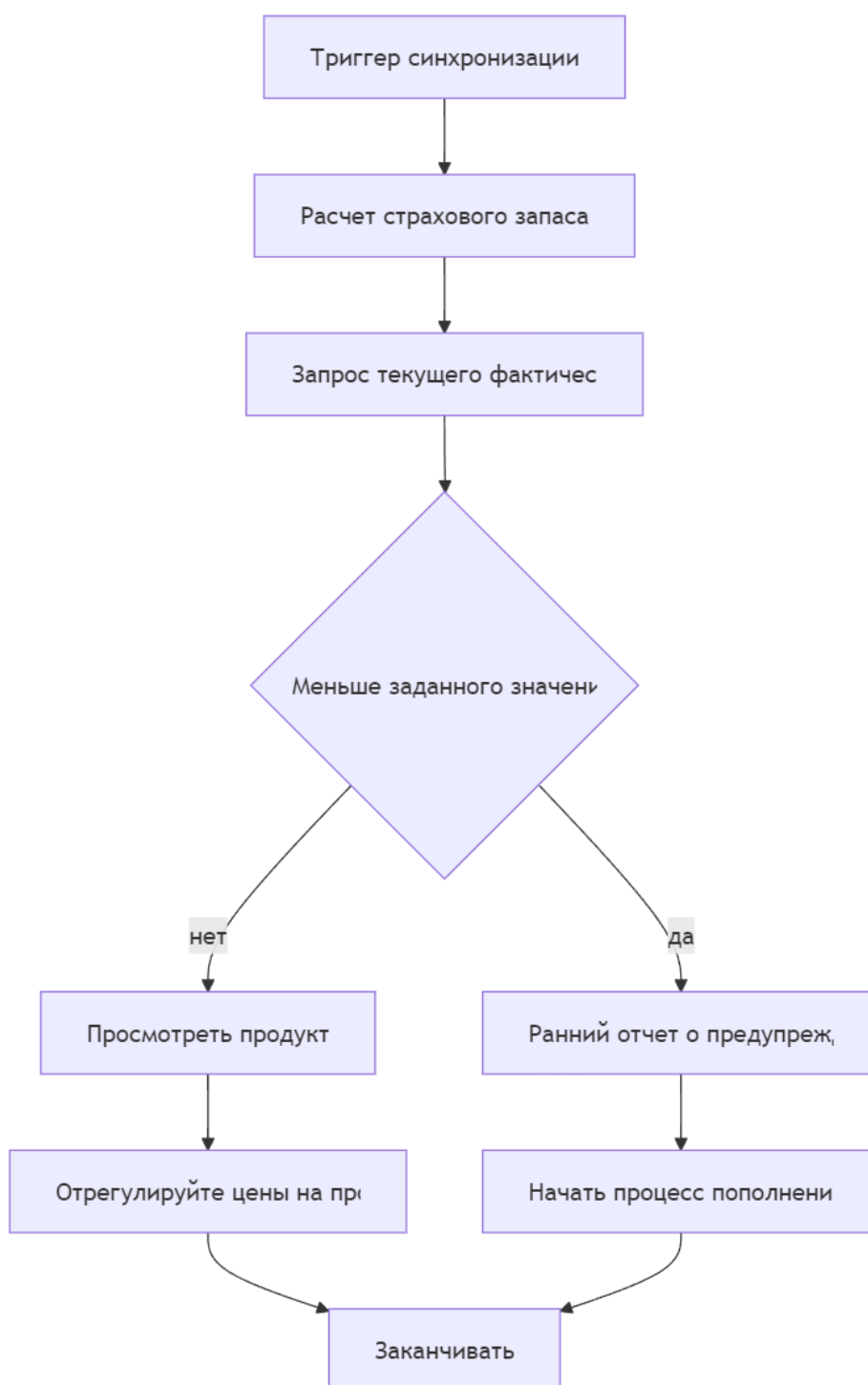


Рисунок 8 – Схема рабочего процесса модуля предупреждения об инвентаризации

Функциональная конструкция модуля: для лучшей координации взаимодействия между различными функциями системы и уменьшения взаимосвязи

между модулями в системе система управления продуктовым магазином разделена на шесть основных модулей, каждый модуль разделен на несколько подмодулей, а также определены конкретные взаимосвязи между модулями. Это можно представить в виде функциональной схемы. Функциональная схема каждого модуля системы управления продуктовым магазином представлена на рисунке 9.

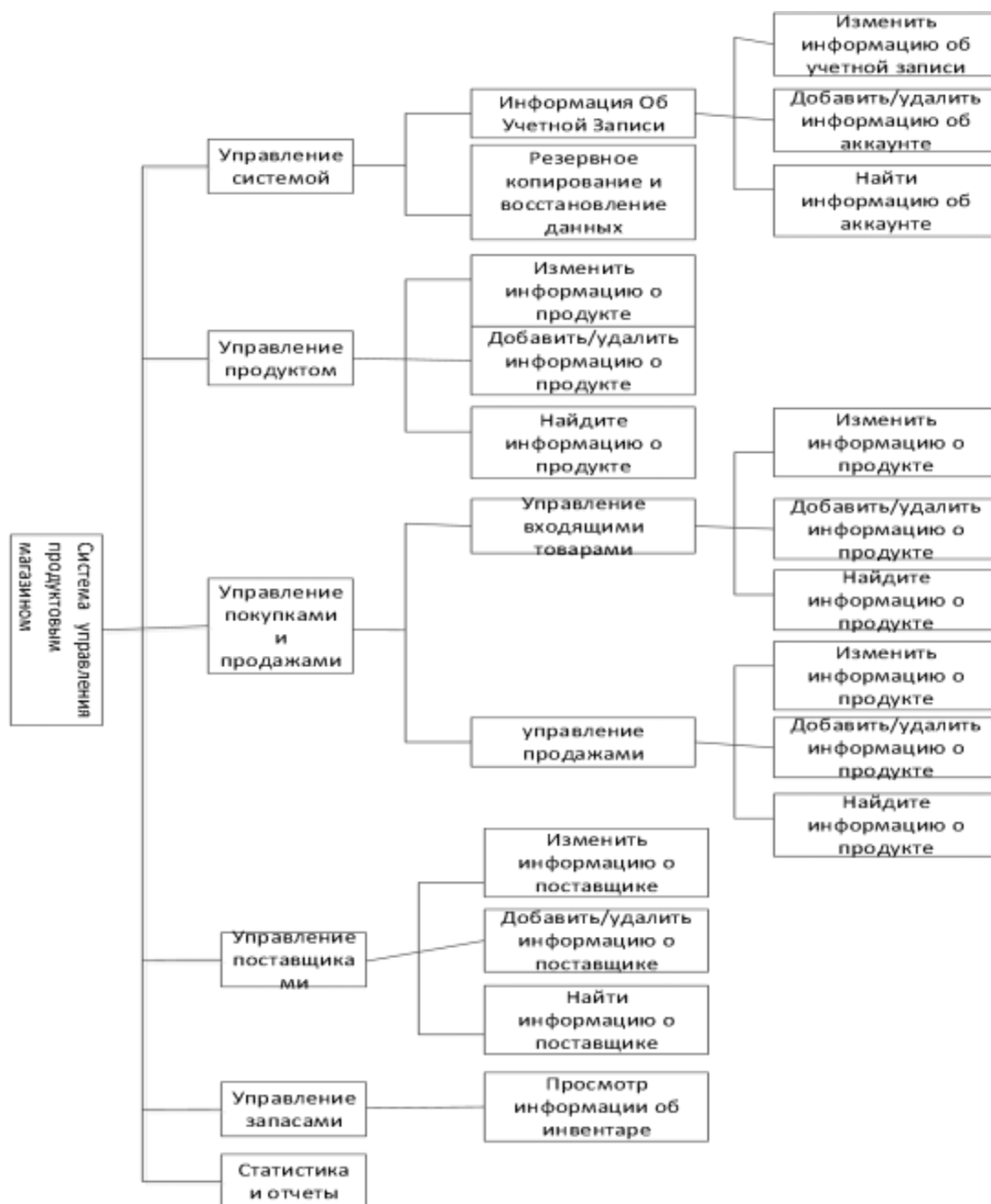


Рисунок 9 – Решение системного функционального модуля

Проектирование базы данных. Система управления продуктовым магазином должна хранить в базе данных информацию о продуктах и поставщиках.

Эта система использует базу данных MySQL. Ниже представлен подробный проект базы данных. Сущность менеджера показана на рисунке 10.



Рисунок 10 – Схема объекта «Администратор»

Товарная сущность представлена на рисунке 11.



Рисунок 11 – Диаграмма сущности продукта

Организация-поставщик показана на рисунке 12.



Рисунок 12 – Диаграмма сущности поставщика
Складское предприятие показано на рисунке 13.



Рисунок 13 – Существующая диаграмма складирования

Объект складского возврата показан на рисунке 14.



Рисунок 14 – Схема объекта складского возврата

Физическая диаграмма возвратов клиентов представлена на рисунке 15.



Рисунок 15 – Физическая схема возврата клиента

2.2 Программные платформы для разработки

Техническая архитектура системы на основе Spring Boot глубоко адаптирована к физическим сценариям розничной торговли. Выбор технологий тесно связан с бизнес-логикой, чтобы сформировать легкое решение с низкими затратами на эксплуатацию и обслуживание и высокой доступностью.

Spring Boot служит в качестве основного фреймворка в системе, проходящего через каждое звено от управления данными до бизнес-логики. Его легкие функции и готовые возможности глубоко отвечают потребностям сценариев физической розничной торговли. Система использует сервер Tomcat, встроенный в Spring Boot, для автономного развертывания. Службу можно запустить, запустив упакованный JAR-файл на кассовом компьютере без настройки профессионального промежуточного ПО. Благодаря автоматизированному механизму настройки система быстро интегрирует встроенную базу данных SQLite, сохраняет данные о продуктах, инвентаризационные записи и информацию о транзакциях в одном файле на локальном жестком диске и реализует объектно-ориентированные операции с базой данных в сочетании с JPA (Java Persistence API). Сохранение осуществляется с помощью метода `save()` интерфейса `ItemRepository`, без необходимости вручную писать операторы SQL на протяжении всего процесса.

Модуль предупреждения о состоянии запасов использует функцию пла-

нирования запланированных задач Spring Boot (аннотация `@Scheduled`) для запуска динамического расчета резервного запаса каждые 30 минут. Алгоритм собирает данные о продажах за последние 72 часа на основе скользящего окна, удаляет аномальные пики и генерирует эластичные значения рекомендаций по пополнению. Когда уровень запасов падает ниже порогового значения, система асинхронно отправляет оповещения WeChat через механизм публикации событий Spring (`ApplicationEventPublisher`), чтобы избежать блокировки основного потока кассира. Срок годности товаров обрабатывается путем бесшовной интеграции Spring Boot с API времени Java 8 для точного расчета оставшегося срока годности партий товаров. После запуска стратегии градиентного продвижения кэшированные ценники продуктов обновляются через `RedisTemplate`, чтобы гарантировать, что информация о продвижении вступит в силу в режиме реального времени в конце оформления заказа.

В дизайне уровня взаимодействия контроллер RESTful Spring Boot (`@RestController`) обеспечивает эффективную поддержку API для апплетов WeChat. Например, при сканировании кода на кассе апплет отправляет штрих-код продукта на бэкэнд через запрос HTTP POST. Spring Boot использует библиотеку Jackson для завершения преобразования между данными JSON и объектами Java в течение 5 миллисекунд, вызывает службу инвентаризации для выполнения атомарных операций вычета и синхронно обновляет отображение инвентаризации в реальном времени на интерфейсе кассы с помощью адаптивного программирования (`WebClient`). Для сценариев с высоким уровнем параллелизма конфигурация пула потоков Spring Boot (`ThreadPoolTaskExecutor`) гарантирует одновременную обработку более 50 запросов на сканирование кода, что позволяет избежать зависаний системы.

По сравнению с традиционными фреймворками разработки основная ценность Spring Boot в этой системе заключается в возможности расширения с «нулевым вмешательством» – когда требуется новый функциональный модуль (например, онлайн-сверка для поставщиков), вам нужно только добавить соот-

ветствующий Spring Bean, чтобы интегрировать его в существующую архитектуру, без необходимости перестраивать базовый код. Например, при интеграции функции оплаты WeChat API оплаты вызывается через RestTemplate, а механизм внедрения зависимостей Spring Boot используется для динамической загрузки параметров конфигурации (идентификатор продавца, ключ API) из application.yml, чтобы гарантировать, что конфиденциальная информация отделена от бизнес-логики. Такая конструкция позволяет системе удовлетворять текущим потребностям небольших магазинов в низкостратной эксплуатации и обслуживании, а также сохранять техническую гибкость для будущих модернизаций модели управления сетью.

Vue.js выступает в качестве основного интерфейса системы, обеспечивая эффективное взаимодействие с пользователем посредством адаптивной привязки данных и проектирования на основе компонентов. Система использует комбинированный API Vue 3 для создания одностраничного приложения (SPA), которое легко подключается к серверной службе Spring Boot через RESTful API. В сценарии управления товарами при сканировании и вводе товаров через визуальный интерфейс на базе Vue система в режиме реального времени вызывает внутренний интерфейс для проверки действительности штрихкода, автоматически заполняет наименование товара, спецификацию и другие поля с использованием двусторонней привязки v-model и динамически рассчитывает рекомендуемую розничную цену на основе рассчитанного атрибута (Computed).

Адаптивная система Vue тесно взаимодействует с модулем WebSocket Spring Boot для обеспечения синхронизации отсканированных данных в реальном времени. Каждый раз, когда сканируется штрих-код продукта, компонент Vue отправляет запрос на бэкэнд через axios для проверки инвентаря. После получения цены общая сумма корзины покупок немедленно обновляется, и запускается анимация перехода номера инвентаря (используя <transition> для обертывания элемента DOM для достижения постепенной анимации).

Element UI служит в качестве базовой интерфейсной структуры бэкэнд-

управления в небольших системах управления продуктовыми магазинами. Благодаря своей высокоинкапсулированной и настраиваемой библиотеке компонентов он преобразует сложные бизнес-операции в интуитивно понятные графические взаимодействия. Система использует контейнер макета Element UI (el-container) для создания адаптивного фона управления. Левое навигационное меню (el-menu) динамически загружает входы модулей, такие как управление продуктами, мониторинг запасов и обработка заказов. Основная область контента справа использует дизайн на основе карточек (el-card) для переноса бизнес-данных. В сценарии отображения данных о товарах компонент el-table реализует высокопроизводительный рендеринг таблиц и поддерживает мгновенный поиск и сортировку десятков тысяч списков товаров.

Система проверки Element UI глубоко применяется в сценариях взаимодействия с формами. Например, когда товары помещаются на хранение, el-form-item, встроенный в el-form, определяет составные правила проверки через атрибут rules. Для многошаговых операций (например, продление контракта с поставщиком) компонент навигации el-steps используется для декомпозиции процесса. Текущий статус шага управляется глобально через Vuex, и прогресс может быть восстановлен, даже если браузер был случайно закрыт.

MySQL используется в качестве основного движка данных в системе управления небольшим продуктовым магазином. Его сильные транзакционные характеристики и высокие возможности параллельной обработки поддерживают стабильную работу модулей, таких как товары, инвентарь и заказы. Система использует механизм хранения InnoDB для обеспечения целостности транзакций ACID. Например, в сценарии кассового чека такие операции, как сканирование кода для вычета инвентаря, создание заказов на продажу и обновление баллов участников, связываются как атомарные транзакции с помощью аннотации @Transactional, гарантируя отсутствие несоответствий между данными инвентаря и финансовыми записями даже в случае внезапного отключения электроэнергии. Для высокочастотных операций запроса уникальный индекс устанавливается в поле штрих-кода таблицы продуктов с помощью

стратегии оптимизации индекса B+ tree, так что время ответа на запрос одного продукта контролируется в пределах 10 миллисекунд. В то же время покрывающий индекс используется для сокращения запроса к задней таблице и повышения эффективности выполнения многофакторного скрининга (например, извлечение по категории + статусу инвентаря).

Интеграция с Spring Boot реализует сопоставление ORM через JPA. В классе сущности аннотация `@DynamicUpdate` используется для обновления только измененных полей, чтобы снизить нагрузку SQL. Для сложных бизнес-запросов (например, для расчета ежемесячной тенденции продаж определенного продукта) оптимизированные собственные операторы SQL записываются с помощью аннотации `@Query`. В ответ на потребности расширения магазина система резервирует интерфейс сегментирования базы данных и таблиц – когда количество записей о товарах в одном магазине превышает 500 000, его можно плавно переключить на ShardingSphere для достижения горизонтального разделения без необходимости перестраивать существующий бизнес-код.

IntelliJ IDEA используется в качестве основной интегрированной среды разработки (IDE) при разработке этой системы. Она глубоко интегрирована в каждое звено конструкции технологического стека. Благодаря интеллектуальной помощи в кодировании и интеграции цепочки инструментов полного стека она значительно повышает эффективность разработки и качество кода. Его глубокая оптимизация для экосистемы Java делает создание внутренних служб Spring Boot еще более эффективным – при создании проекта напрямую выберите шаблон Spring Initializr, чтобы автоматически сгенерировать файл Maven POM, содержащий необходимые зависимости, такие как `spring-boot-starter-data-jpa` и `spring-boot-starter-web`, а также используйте встроенный инструмент анализа зависимостей для выявления конфликтов версий в режиме реального времени (например, конфигурации совместимости MyBatis и JPA). При написании класса сущности модуля управления продуктом генератор кода IDE (`Alt+Insert`) генерирует аннотации JPA (такие как `@Entity`, `@Table`) одним

щелчком мыши. В сочетании с плагином Lombok аннотация `@Data` автоматически вставляется для устранения избыточного кода Getter/Setter, что повышает эффективность сопоставления сущностей более чем на 60 %.

На уровне разработки front-end полная поддержка API композиции Vue 3 достигается путем установки подключаемого модуля Vue.js: при написании компонента сканирования QR-кода интерфейса кассира IDE интеллектуально запрашивает инструкции двусторонней привязки `v-model` и функции API композиции (такие как `ref()`, `computed()`), а также автоматически запускает стандартную проверку кода ESLint при сохранении файла для перехвата необъявленных переменных или некорректных вызовов методов. Библиотека компонентов Element UI ускоряет разработку с помощью встроенной библиотеки фрагментов кода. Конфигурация атрибутов (например, `data="itemList"`) выполняется с помощью визуальной формы. На этапе отладки отладчик JavaScript IDE бесшовно интегрируется с DevTools браузера Chrome, что позволяет устанавливать точки останова для фиксации аномалий потока данных в логике обработки событий сканирования кода.

На этапе тестирования интеграции системы инструмент HTTP-клиента IDE заменил Postman в качестве первого выбора для отладки API: напишите запрос POST `http://localhost:8080/api/checkout` в файле `rest-api.http`, нажмите кнопку «Выполнить», чтобы напрямую отправить данные JSON, содержащие массив штрихкодов продуктов; результат ответа автоматически форматируется и отображается, а также поддерживается сравнение с историческими запросами для анализа колебаний производительности. Для сложных сценариев, таких как параллельное тестирование вычета инвентаря, мы используем подключаемый модуль JMH (Java Microbenchmark Harness), интегрированный в IDE, для написания контрольных тестов, имитирующих 100 потоков для одновременной инициации запросов сканирования кода и точного определения узкого места пропускной способности в рамках оптимистичного механизма блокировки JPA.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРЕДЛАГАЕМОГО АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ

3.1 Основные этапы практической разработки программного продукта

Разработка системы управления продуктовым магазином в основном состоит из четырех этапов: анализ спроса, выбор технологии, проектирование, разработка системы, тестирование и развертывание.

Основные функциональные требования. Управление информацией о товаре: Поддерживает такие поля, как название товара, штрих-код, категория (продукты питания/товары первой необходимости и т. д.), закупочная цена, продажная цена, срок годности, пороговое значение запасов и т. д., обновляет запасы в режиме реального времени, поддерживает преобразование нескольких единиц (например, коробок/бутылок/штук) и автоматически предупреждает о низком уровне запасов или истекающем сроке годности товаров. Запрос продукта, быстрый поиск по названию, категории, штрих-коду, поддержка пакетного изменения цен или информации об акциях.

Управление продажами. Функция кассира, сканирование кода/ручной ввод товаров, расчет общей стоимости, поддержка наличных и электронных платежей. Регистрируйте время, сведения о продукте и способ оплаты каждой транзакции, а также осуществляйте операции возврата и обмена.

Управление закупками и поставщиками. Регистрируйте название поставщика, контактную информацию, исторические записи о закупках и оценивайте стабильность поставок. Формируйте предложения по закупкам на основе предупреждений о состоянии запасов, поддерживайте ручное создание заказов и отслеживайте статус прибытия. Обновляйте инвентарь после приемки товаров, регистрируйте номер партии и срок годности, а также поддерживайте управление многопартийным запасом.

Финансы и отчеты: Статистика продаж, ежедневные/ еженедельные/

ежемесячные продажи, валовая прибыль и рейтинги самых продаваемых продуктов.

Анализ затрат, расходы на закупку, стоимость инвентаря, отчет о прибыли (по продукту/категории/периоду времени), экспорт данных, создание отчетов Excel, поддержка финансовой сверки.

Управление системой: данные автоматически резервируются в облаке/локально, поддерживается восстановление данных; конфиденциальная информация шифруется и сохраняется.

Техническая осуществимость: используется Spring Boot+ Vue.js+ MySQL. Существует большое количество зрелых решений для системных функций (таких как управление запасами, кассовый аппарат и управление участниками). Техническая сложность реализации относительно низкая. Интеграционные решения аппаратных устройств, таких как сканеры штрихкодов и принтеры чеков, с программными системами стандартизированы, а риски разработки контролируемы.

Операционная осуществимость: легко обучать продавцов магазина, а интерфейс кассира похож на POS-систему супермаркета, которая соответствует обычным рабочим привычкам. Операционный статус можно быстро понять с помощью визуальных отчетов (например, самые продаваемые продукты и тенденции прибыли).

Что касается аппаратного обеспечения, обычный домашний компьютер может поддерживать разработку и последующее использование этой системы. Программное обеспечение использует IDEA, а база данных – MySQL. Разработчик много раз использовал оба программного обеспечения в студенческой жизни для соединения базы данных и проекта. Его можно беспрепятственно использовать. SpringBoot Фреймворк прост в использовании, поэтому разработка системы управления продуктовым магазином осуществима с технической точки зрения.

Экономическая целесообразность: Разработка системы управления про-

дуктовым магазином заняла около 40 дней. Принимая во внимание ежемесячную зарплату инженеров-программистов в районе Хэйлунцзян, она составляет 5000 юаней, а плата за Интернет в течение 40 дней составляет 98 юаней. В этой системе используется инструмент разработки IDEA, база данных MySQL и среда SpringBoot, которые в настоящее время популярны и являются бесплатными и с открытым исходным кодом, поэтому в процессе разработки стоимость проекта составляет около 6000 юаней, а зарплата сотрудников продуктового магазина – около 3000 юаней. После разработки системы одна зарплата сотрудника можно сэкономить, а ежемесячная чистая прибыль продуктового магазина составляет около 3500 юаней, поэтому предполагается, что после ввода этой системы в эксплуатацию средства, использованные для развития, могут быть возвращены в течение примерно 2-3 месяцев. С экономической точки зрения вполне возможно разработать систему управления продуктовым магазином.

Требования к безопасности системы и возможность соблюдения правовых норм: Безопасность очень важна для системы. Только авторизованные пользователи могут получать доступ к информации и изменять ее. Чтобы предотвратить утечку информации, создайте механизм безопасности. Даже если система где-то не может нажать какую-то функцию, она все равно дает сбой. Вероятно, все будет хорошо работать.

Хранение информации должно соответствовать Закону о защите персональных данных. Система должна поддерживать всплывающие окна для соглашений о конфиденциальности и анонимизации данных. Платежный интерфейс должен быть подключен к лицензированному учреждению (например, официальному поставщику платежных услуг WeChat), чтобы избежать риска двойного клиринга. Записи о продажах должны соответствовать требованиям налогового аудита (например, сохранение потоков транзакций более 5 лет). Система должна генерировать стандартные финансовые ваучеры. Электронные квитанции должны соответствовать соответствующим положениям закона об электронной подписи.

Выбор технологии является ключевым звеном в разработке информационной системы, напрямую влияющим на производительность, ремонтпригодность и масштабируемость системы. В ответ на потребности систем управления продуктовыми магазинами (таких как управление продукцией, обработка заказов, мониторинг запасов и адаптация к нескольким терминалам) требуется комплексная оценка на основе таких измерений, как зрелость технологий, эффективность разработки и экономическая эффективность, а также подробное обсуждение теоретических основ выбора технологий, сравнительный анализ технологий-кандидатов и процесс принятия окончательного решения.

Выбор технологий следует принципу баланса, обусловленного спросом и экономически эффективного. Аналитический иерархический процесс (АНР) используется для многомерной оценки. Системные требования разлагаются на функциональные (например, обновление инвентаря в реальном времени) и нефункциональные (например, поддержка высокой степени параллелизма) показатели. На основе отраслевой практики и литературных исследований выбирается основной технологический стек. Установите пять параметров оценки: производительность, эффективность разработки, поддержка сообщества, безопасность и стоимость, а также назначьте веса (таблица 1).

Таблица 1 – Таблица распределения веса

Оценка размеров	Вес	Описания
1	2	3
Производительность	0.3	Скорость отклика, вычислительная мощность
Эффективность разработки	0.25	Кривая обучения, совершенство инструмента
Поддержка сообщества	0.2	Качество документа, активность экосистемы
Безопасность	0.15	Шифрование данных

Стоимость	0.1	Расходы на потребление
-----------	-----	------------------------

Сравнительный анализ технологий-кандидатов и причины их выбора.
Выбор фреймворка бэкэнда: Spring Boot, Django.

Spring Boot (JAVA): Оптимизация пула потоков на основе JVM и управление пулом соединений (например, HikariCP), которые могут поддерживать сценарии с высокой степенью параллелизма – более 500 заказов в секунду. Аннотация `@Transactional` объединена с движком MySQL InnoDB для обеспечения атомарной операции вычета запасов и оплаты заказов. Spring Security обеспечивает комплексное управление разрешениями RBAC, а Spring Cloud поддерживает расширение микросервисов.

Однако конфигурация сложна и требует знакомства с экосистемой Spring (например, JPA и Feign), поэтому первоначальные затраты на обучение высоки.

Django (Python): встроенный административный бэкэнд и ORM, поддерживает быструю генерацию CRUD-интерфейсов, подходит для быстрой итерации небольших и средних проектов, синтаксис Python лаконичен, подходит для обработки задач с интенсивным вводом-выводом (например, асинхронных уведомлений о заказах).

Однако из-за ограничений блокировки GIL производительность многопоточности низкая, и сложно поддерживать сценарии с высоким параллелизмом (например, мгновенные продажи).

В конечном итоге Spring Boot стал окончательным выбором из-за следующих основных преимуществ: объем заказов резко возрос в период рекламной акции продуктового магазина, управление пулом потоков Spring Boot оптимизация пула соединений могут обеспечить стабильность системы и сформировать более удобную связь с выбранным на фронте Vue.js.

Выбор фреймворка для фронтенда: Vue.js, React.

Vue.js: Прогрессивный фреймворк: функции можно интегрировать постепенно, а компоненты в виде отдельных файлов (SFC) улучшают читаемость кода. Двухнаправленная привязка данных упрощает логику обновления инвентаря в реальном времени и повышает эффективность разработки на 40%.

Поддержка нескольких терминалов: с помощью Uni-App его можно скомпилировать в апплет WeChat, H5 и приложение, что снижает стоимость разработки для нескольких терминалов.

Однако в крупных проектах необходимо строго регулировать управление состоянием Vuex, в противном случае это легко приведет к избыточности кода.

React: Virtual DOM оптимизирует производительность рендеринга, Ant Design Pro предоставляет фоновые шаблоны корпоративного уровня.

Однако кривая обучения синтаксису SX и управлению состоянием Redux довольно крутая, а небольшие и средние проекты склонны к излишней сложности проектирования.

Наконец, в качестве front-end был выбран Vue.js. Адаптивная привязка данных Vue упрощает логику обновления инвентаря в реальном времени.

При расширении мини-программ в будущем 80 % кодов Vue можно будет повторно использовать через Uni-App, что позволит сократить затраты на разработку многотерминальных приложений. А сочетание Vue.js + Element UI обеспечивает наилучшие показатели эффективности разработки и функционального покрытия. Выбор библиотеки компонентов пользовательского интерфейса: Element UI, Ant Design. Element UI: из коробки он предоставляет 90% внутренних компонентов, таких как формы (`<el-form>`) и таблицы (`<el-table>`), сокращает кодирование на 30 % и имеет китайский языковой пакет и формат даты по умолчанию, что снижает затраты на настройку. Однако расширенные диаграммы полагаются на сторонние библиотеки, такие как ECharts. Ant Design: шаблон ProLayout поддерживает быстрое создание сложного фона. Однако для настройки темы требуется изменение переменных Less, что не очень удобно для новичков.

В конце концов Element UI победил благодаря своей глубокой интеграции с Vue.js и преимуществам локализации. Готовые компоненты Element UI (такие как загрузка и выбор даты) могут напрямую удовлетворять потребности управления запасами. Используйте v-model для достижения синхронизации в

реальном времени между данными формы и API бэкенда, чтобы оптимизировать взаимодействие с пользователем.

Выбор базы данных между MySQL и PostgreSQL. MySQL: благодаря блокировке на уровне строк и уровню изоляции транзакций движка InnoDB гарантируется согласованность данных о заказах и инвентаре, достигается автоматическое резервное копирование и разделение чтения и записи, а также сокращаются расходы на эксплуатацию и обслуживание. Однако сложные аналитические запросы (например, прогнозирование тенденций продаж) требуют использования дополнительных инструментов OLAP. PostgreSQL: поддерживает поля JSONB и геопространственные индексы, подходит для хранения неструктурированных данных. Однако затраты на эксплуатацию и обслуживание на уровне предприятия высоки, и небольшим группам сложно в полной мере использовать свои функции.

Наконец, мы выбрали MySQL, поскольку он больше подходит для сценариев продуктового магазина с точки зрения безопасности транзакций и удобства эксплуатации и обслуживания. Оплата заказа и вычет запасов должны строго соответствовать ACID, а пессимистический механизм блокировки MySQL (такой как SELECT FOR UPDATE) позволяет избежать перепродажи. Такие инструменты, как Persona Toolkit, поддерживают онлайн-изменения DDL, не влияя на доступность системы в рабочее время.

Окончательный технологический стек и преимущества интеграции показаны в таблице 2.

Таблице 2 – Технологии и преимущества

Технологический стек	Краткое изложение причин выбора
1	2
Spring Boot	Высокая производительность, масштабируемость корпоративного уровня и интеграция с Vue.js

Vue.js	Прогрессивная структура снижает порог разработки и взаимодействует с Uni-App для достижения «единовременной разработки, многотерминального покрытия».
--------	---

Продолжение таблицы 2

Element UI	Система управления бэкэндом охватывает все компоненты, сокращая цикл разработки фронтэнда на 30%, и имеет полную поддержку локализации.
MySQL	Безопасность транзакций и высокая доступность защищают основные бизнес-данные, а в сочетании с кэшем Redis производительность запросов к инвентарю повышается до уровня миллисекунд.

Схема выбора технологий обеспечивает баланс между эффективностью разработки, производительностью и требованиями к долгосрочной масштабируемости посредством структурированной демонстрации и количественной оценки, что делает выбранные технологии более подходящими для разработки этой системы.

Фаза проектирования и разработки системы. Проектирование архитектуры системы определяет техническую основу системы и ее многоуровневую структуру для обеспечения масштабируемости и стабильности.

Техническая архитектура использует режим разделения front-end и back-end, используя Vue.js для front-end, Spring Boot для back-end и MySQL для базы данных.

Многослойная архитектура системы:

- уровень представления отвечает за взаимодействие с пользователем и реализует пользовательский интерфейс для различных ролей, включая кассира, персонал административного уровня и мобильных пользователей. Интерфейс адаптирован под различные устройства, обеспечивая интуитивно понятный доступ к функциям системы как на стационарных терминалах, так и на мобильных платформах;

- уровень бизнес-логики реализует основные функциональные процессы, включая управление ассортиментом товаров, расчёт и прогнозирование

запасов, проведение транзакций, применение скидок и акций, а также обработку возвратов. Вся логика централизована и отделена от пользовательского интерфейса, что упрощает масштабирование и сопровождение системы;

– уровень доступа к данным инкапсулирует работу с базой данных, обеспечивая безопасный и эффективный доступ к информации. Поддерживаются механизмы управления транзакциями, репликация и резервное копирование, а также внедрено кэширование часто используемых данных для повышения производительности и снижения нагрузки на хранилище;

Архитектура развертывания может быть реализована в двух вариантах: в виде автономного решения на локальном сервере – для точек продаж с ограниченным интернет-доступом, либо как облегчённый облачный сервис с использованием ресурсов, таких как Alibaba Cloud ECS и RDS – для обеспечения гибкости, масштабируемости и высокой доступности. Подключение через облачную инфраструктуру также позволяет централизованно управлять обновлениями и безопасностью, сокращая затраты на обслуживание. Архитектурная схема представлена на рисунке 16.

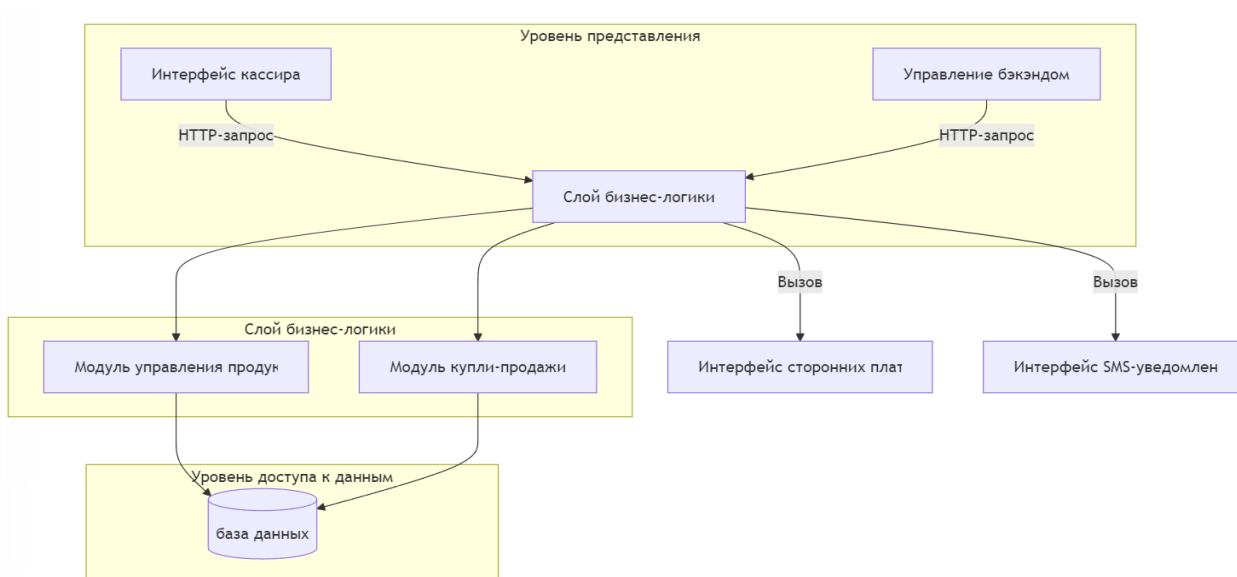


Рисунок 16 – Схема архитектуры системы

Разметка основных функциональных модулей и уточнение логики взаимодействия позволяют структурировать систему таким образом, чтобы каждый

модуль выполнял четко определённые задачи, обеспечивая при этом взаимосвязь с другими компонентами. В текущей архитектуре система разделена на пять ключевых модулей: управление закупками, управление продажами, управление заказами, управление продуктами и модуль раннего оповещения.

В рамках процесса закупки в информационной системе формируется заказ, привязанный к конкретному поставщику. Пользователь осуществляет выбор необходимых товарных позиций, указывает их количество, закупочную цену, а также, при необходимости, – планируемую дату поставки и условия оплаты. После этого заказ передаётся на этап согласования, включающий проверку соответствия установленным бюджетным ограничениям, лимитам закупок, а также требованиям к аккредитованным поставщикам.

По завершении процедуры согласования система автоматически генерирует приходную накладную, на основании которой осуществляется приёмка товаров на склад. В результате приёмки инвентарные данные обновляются в системе в режиме реального времени, обеспечивая актуальность учёта материальных ценностей. Предусмотрена возможность регистрации частичных поставок, что позволяет фиксировать отклонения от согласованного графика и объёма поставки.

Функционал системы включает ведение детализированной истории взаимодействий с каждым поставщиком. В эту информацию входят данные о фактических сроках выполнения поставок, точности исполнения количественных и качественных параметров заказов, а также соблюдении договорных условий. На основе указанных показателей система формирует интегральную оценку надёжности поставщика, которая может быть использована для принятия обоснованных управленческих решений в последующих закупочных процедурах.

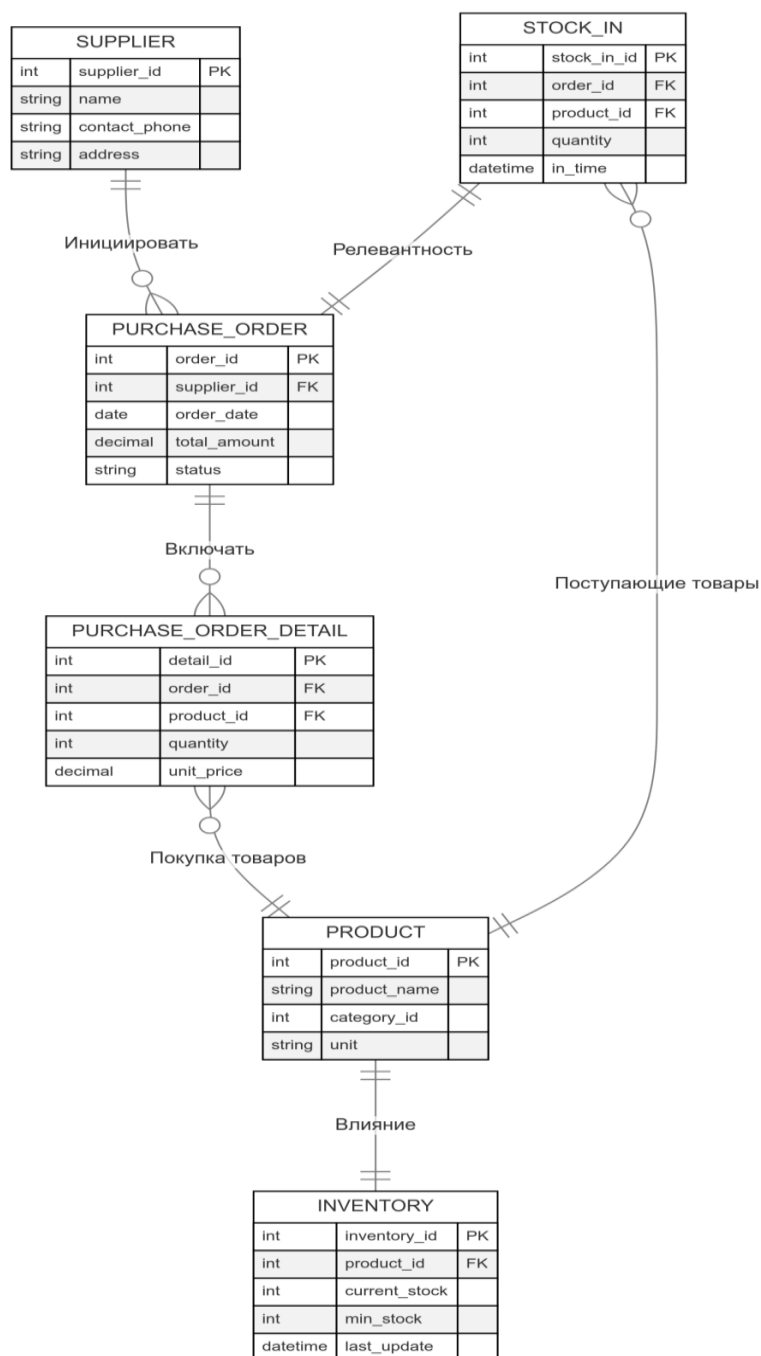
При достижении минимального допустимого уровня товарных остатков система инициирует автоматическое формирование предварительного предложения на повторную закупку. Данный механизм учитывает прогнозируемый спрос, сезонные колебания, динамику потребления, а также исторические дан-

ные о предыдущих закупках, что позволяет обеспечивать устойчивость снабжения и предотвращать возникновение дефицита.

Система также предусматривает возможность интеграции с внешними информационными платформами (например, модулями бухгалтерского учёта или ERP-системами), поддержку механизмов оповещений и формирование аналитических отчётов по ключевым показателям эффективности закупочной деятельности.

В модуле предусмотрена интеграция с системой бухгалтерского учёта для автоматического отражения движения товаров и формирования отчётности по закупочной деятельности. Все действия фиксируются в журнале аудита с возможностью последующего анализа. Проект ER-диаграммы модуля управления закупками представлен на рисунке 17.

Модуль управления закупками ER-диаграмма



Рисунке 17 – Схема ER модуля управления закупками

Ниже приводится описание объектов, как показано на рисунке 3.

Таблица 3 – Описание объекта

Имя объекта	Описание поля
1	2
SUPPLIER (Поставщик)	Идентификатор поставщика, имя, контактный номер, адрес

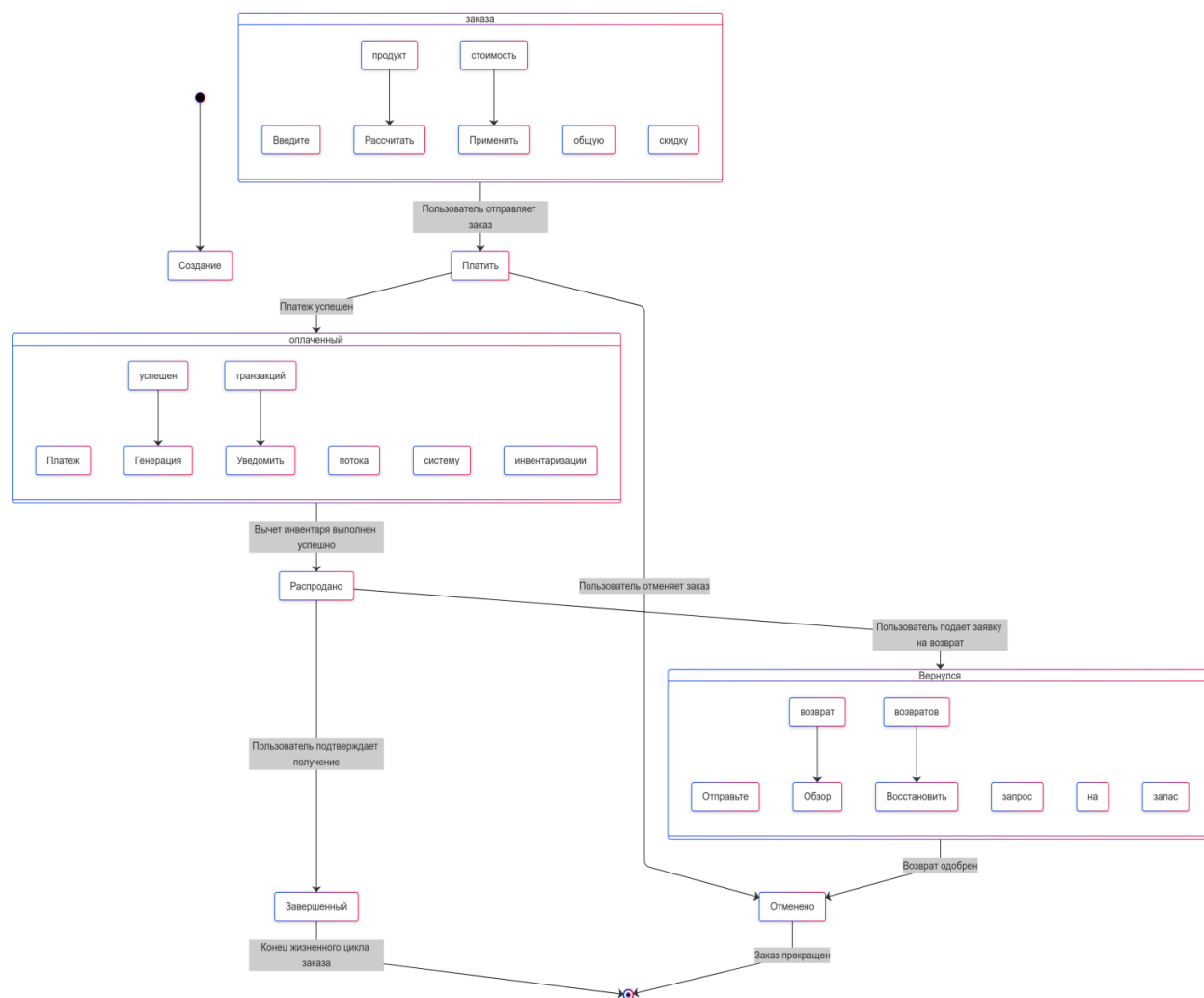
PURCHASE_ORDER (заказ на покупку)	Идентификатор заказа, идентификатор поставщика, дата заказа, общая сумма, статус
PURCHASE_ORDER_DETAIL (Детали покупки)	Идентификатор детали, идентификатор заказа, идентификатор продукта (внешний ключ), количество покупки, цена за единицу.
PRODUCT (товар)	Идентификатор продукта, название, идентификатор категории, единица измерения
STOCK_IN (Складские записи)	Идентификатор входящего склада, идентификатор заказа, идентификатор продукта, количество складирования, время складирования
INVENTORY (в наличии)	Идентификатор инвентаря, идентификатор продукта, текущий запас, минимальное значение предупреждения об инвентаре

Модуль продаж осуществляет обработку операций розничной реализации товаров посредством сканирования штрихкодов на кассовом узле. При регистрации каждой продажи система в режиме реального времени обновляет данные об остатках на складе, обеспечивая актуальное отображение товарных запасов. Одновременно с этим автоматически формируются записи о продажах, содержащие информацию о наименовании товара, количестве, цене, дате и времени операции, а также идентификаторе кассира и точки продаж.

Кроме того, модуль обеспечивает проверку корректности сканируемых данных, контроль за применением скидок и акций, а также регистрацию сопутствующих параметров, включая способ оплаты и параметры лояльности покупателя, если применимо. Информация о каждой транзакции может быть использована для последующего анализа потребительского спроса, формирования отчётности, а также расчёта ключевых показателей эффективности (KPI) розничной сети.

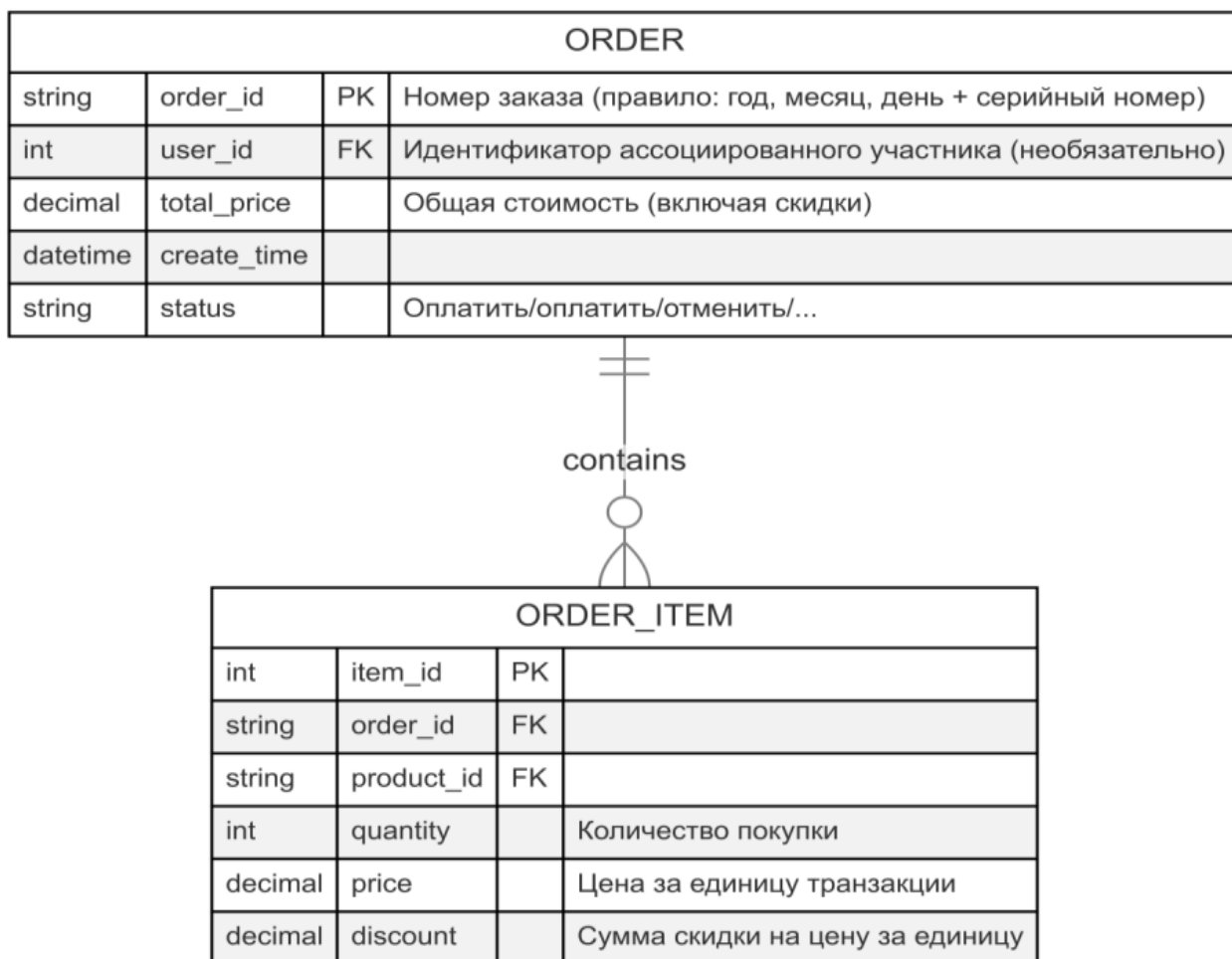
Представленный ниже поток статуса продаж, отражающий жизненный цикл операции от момента сканирования товара до фиксации результата в базе данных, визуализирован с помощью диаграммы состояний на рисунке 18. Диаграмма демонстрирует переходы между состояниями системы, такими как

«Ожидание сканирования», «Обработка позиции», «Расчёт итоговой суммы», «Проведение оплаты» и «Формирование отчётной записи», что позволяет проследить логическую последовательность и внутреннюю структуру процесса продажи.



Рисунке 18 – Процесс статуса продаж

Модуль управления заказами отвечает за обработку всего бизнес-процесса пользователя от размещения заказа до завершения транзакции. Модель ключевых данных показана на рисунке 19.

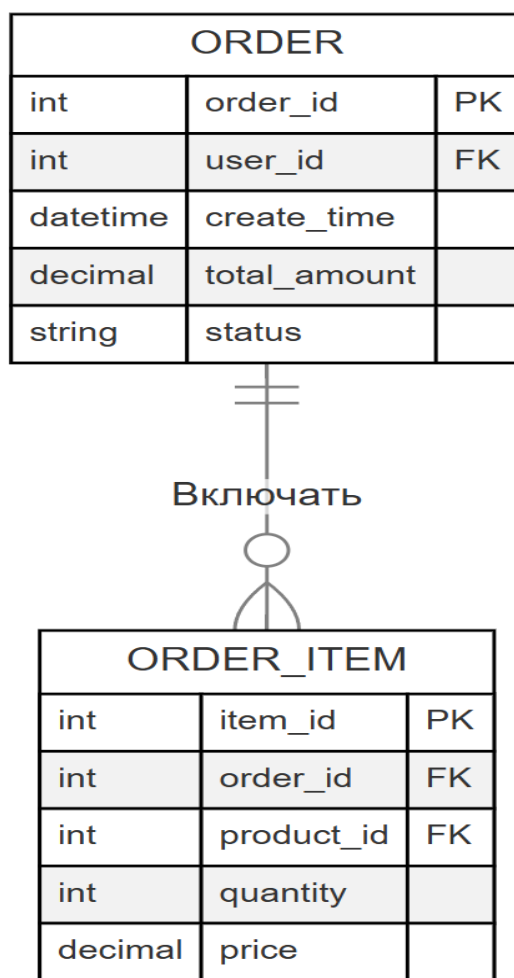


Рисунке 19 – Модель ключевых данных

Модуль управления продуктами отвечает за ввод, редактирование и удаление продуктов. Система поддерживает пакетное редактирование базовой информации о товарах, такой как цена, запасы, вес и т. д., а также может устанавливать многомерные теги поиска для облегчения быстрого поиска товаров. Благодаря интеллектуальной системе классификации товары могут автоматически классифицироваться по таким факторам, как сезоны, узлы продвижения и т. д., повышая эффективность управления.

Система поддерживает многоуровневое управление категориями, позволяя добавлять, редактировать и удалять категории, а также поддерживает загрузку и выгрузку продуктов.

Система поддерживает гибкую настройку цен. Для одного SKU можно установить разные цены, параметры запасов и логистики для удовлетворения потребностей различных сценариев. Диаграмма ER показана на рисунке 20.



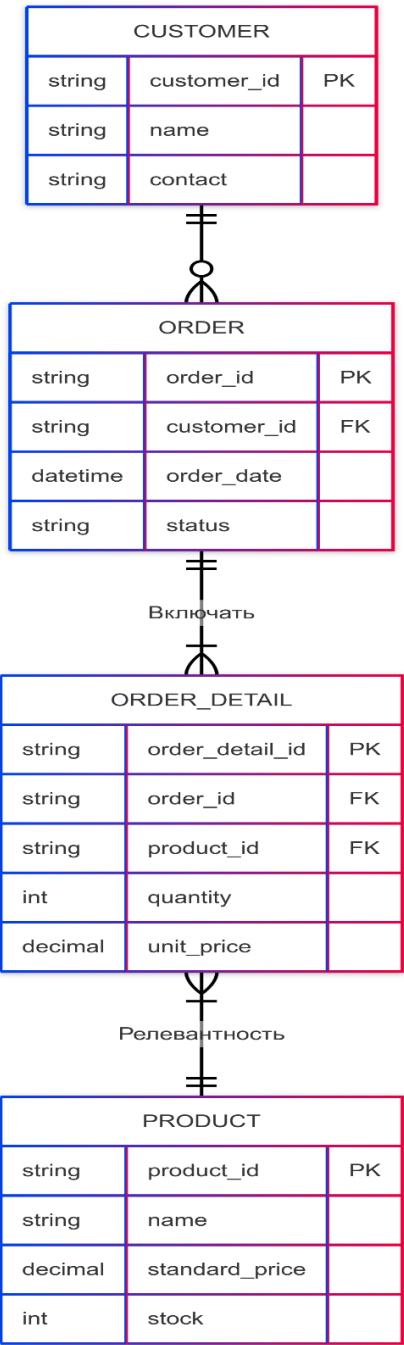
Рисунке 20 – Модуль управления продуктами

Каждый модуль информационной системы логически и функционально взаимосвязан с другими компонентами, что обеспечивает целостность бизнес-процессов и согласованность данных на всех этапах обработки информации. Взаимодействие модулей реализовано посредством унифицированных интерфейсов и общих структур хранения данных, что позволяет обеспечивать высокую степень интеграции между подсистемами, такими как закупки, склад, продажи, финансы и аналитика.

В частности, модуль управления заказами тесно интегрирован с модулем товарного учёта: заказы автоматически ассоциируются с конкретными товарными позициями, на основании которых формируются документы движения, актуализируются складские остатки и осуществляется контроль за выполнением обязательств перед поставщиками и покупателями.

Структура взаимосвязей между заказами и продуктами представлена на

диаграмме в рисунке 21. Данная диаграмма наглядно иллюстрирует логику отношений между сущностями, демонстрируя, каким образом каждый заказ может содержать одну или несколько товарных позиций, а каждая позиция — быть частью различных заказов в рамках разных бизнес-процессов. Такая структура способствует прослеживаемости движения товаров, аналитике закупочной и сбытовой деятельности, а также обеспечивает основу для автоматизированного контроля остатков и планирования потребностей.



Рисунке 21 – Модель «сущность-связь»

Функция модуля раннего предупреждения отслеживает ключевые бизнес показатели в режиме реального времени, запускает ранние уведомления при возникновении отклонений и помогает менеджерам принимать быстрые решения. Типы раннего предупреждения показаны в таблице 4, а машина состояний раннего предупреждения показана на рисунке 22.

Таблице 4 – Тип предупреждения

Тип	Условие триггера	Метод обработки
1	2	3
Предупреждение о низком уровне запасов	Запасы продукции $\leq$ порога страхового запаса	Автоматически формируйте предложения по покупке и уведомляйте менеджеров магазинов о необходимости пополнения запасов.
Предупреждение о переизбытке запасов	Непродаваемый период продукта составляет ≥ 30 дней, а запас составляет ≥ 50 штук.	Запускайте акции или переносите в другие магазины
Немедленное предупреждение опродукте	Оставшиеся дни срока годности продукта ≤ 7 дней	Снижение цен или удаление с полок
Предупреждение о нарушениях продаж	Снижение продаж за один день ≥ 30 % в годовом исчислении или уровень отмены заказов ≥ 15 %	Проверьте качество продукции, стратегию ценообразования или сбоя системы.
Предупреждение об отказе в оплате	5 последовательных неудачных платежей (канал оплаты может быть ненормальным)	Переключитесь на альтернативные каналы оплаты и сообщите техническим специалистам о необходимости расследования.



Рисунке 22 – Государственная машина раннего предупреждения

Ниже приведено подробное введение в модуль управления клиентами в системе управления продуктовым магазином, который разделен на три части, основные функции, приложения для работы с данными и ценность для бизнеса:

Благодаря внедрению микросервисной архитектуры управление клиентами разделено на независимые сервисные модули, а возможности предоставляются извне через шлюз API.

Кластер MySQL хранит основную информацию об участниках (такую как уровень и баллы) и поддерживает большие объемы данных посредством сегментирования. База данных хранит журналы поведения пользователей (например, записи о кликах и покупках) и использует свои высокие характеристики пропускной способности для обработки данных временных рядов. Создайте поисковую систему тегов для поддержки сложных комбинаций тегов

(например, «высокое потребление + предпочтения матерей и детей»).

Вычислительный механизм в режиме реального времени, использующий Flink для обработки потоков поведенческих данных (например, обновлений статуса активности пользователя в режиме реального времени), пакетного анализа исторических данных на основе Spark и генерации долгосрочных тегов (например, годовых итогов потребления).

Управляйте правилами уровня и транзакциями точек, а также обеспечьте параллельную согласованность с помощью распределенных блокировок. Динамически рассчитывайте портреты пользователей и поддерживайте запросы к сторонним системам через API. Управляйте конфигурацией кампании и многоканальными push-уведомлениями, интегрируя SMS, уведомления через приложение и другие методы связи.

Основные функции включают управление системой членства, динамическую корректировку уровней членства, поддержку пользовательских условий повышения уровня (таких как совокупный объем потребления, активность за последние 30 дней), расчет статуса членства на рассвете каждого дня, отправку уведомлений о повышении/понижении уровня (SMS/ Push уведомления через приложение), ограничение максимального количества баллов, зарабатываемых в день, и внесение пользователей в черный список.

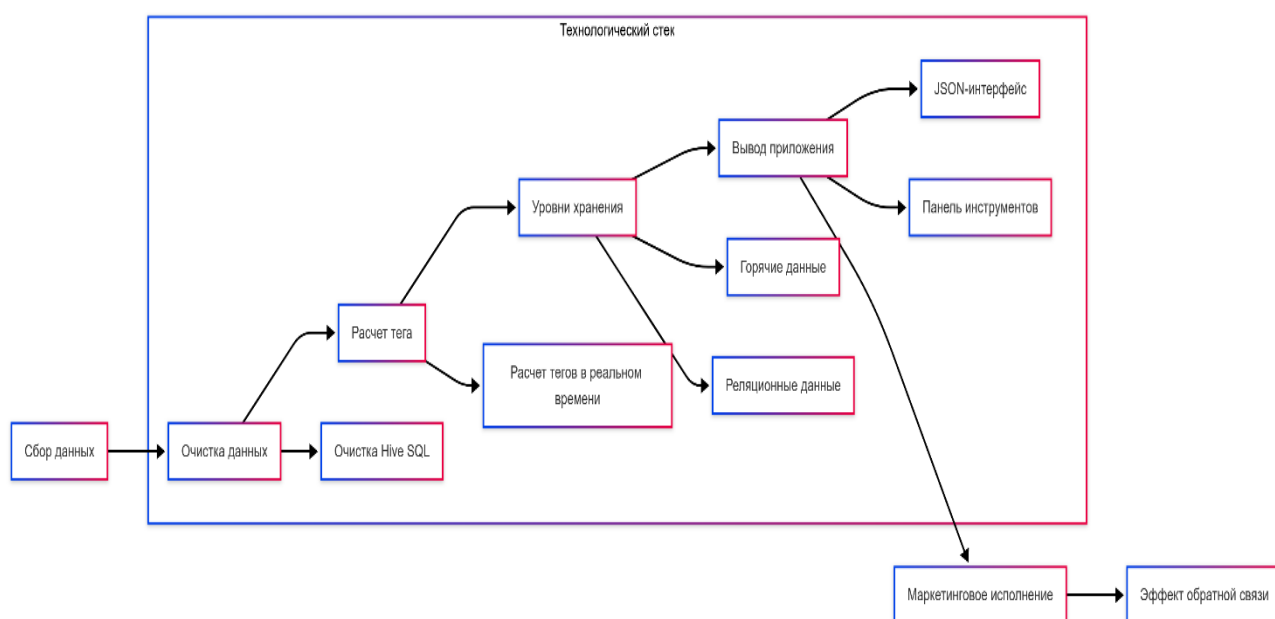
Проводите маркетинг на основе сценариев, включая праздничные акции (товары к празднику Весны), распродажу запасов (целевая распродажа товаров с истекающим сроком годности), поздравления с днем рождения + эксклюзивные купоны и звонки-будильники для пользователей, которые долгое время не заходили в систему.

Проводите маркетинг на основе сценариев, включая праздничные акции (товары к празднику Весны), распродажу запасов (целевая распродажа товаров с истекающим сроком годности), поздравления с днем рождения + эксклюзивные купоны и звонки-будильники для пользователей, которые долгое время не заходили в систему.

Интеллектуальный выбор канала, клиенты с низкой стоимостью используют пакетную рассылку SMS (стоимость $\leq 0,05$ юаня/сообщение), клиенты с высокой стоимостью пользуются эксклюзивными телефонными повторными визитами службы поддержки клиентов.

Используйте иерархическую модель RFM для анализа данных и примените алгоритм Apriori для поиска комбинаций продуктов, комбинирования настроек скидок и оптимизации выкладки товаров на полках.

Типичное принятие решений на основе данных: оптимизация динамического ценообразования, предоставление «купонов на полную скидку» вместо «купонов на скидку» группам клиентов, чувствительным к высоким ценам, предоставление дополнительных услуг с «приоритетной доставкой» группам клиентов с высоким уровнем дохода и проверка клиентов, которые «приобрели продукт А на прошлой неделе», запуск напоминания о повторной покупке после того, как продукт А пополнится в магазине. Схема процесса обработки данных представлена на рисунке 23.



Рисунке 23 – Схема обработки данных

Связано с управлением продажами, историческими данными о потреблении клиентами, оптимизирует выбор рекламной продукции, связано с управлением заказами, адресами высокочастотной доставки, корректирует стратегию

пополнения склада, связано с управлением продуктами, оценкой наиболее продаваемых продуктов, улучшением работы поставщиков методом обратного проталкивания.

Количественная оценка преимуществ, создание платформы данных о клиентах (CDP) на уровне предприятия, поддержка разработки новых продуктов (например, корректировка покупок на основе тенденций поиска «здоровой пищи»), баллы участников и внешние бренды (например, баллы, используемые для погашения купонов Starbucks), повышение лояльности пользователей, контроль рисков и механизм правил, пресекающий ненормальное поведение. Повышение ценности жизненного цикла клиента, основная логика – точный маркетинг, чтобы увеличить повторные покупки и в конечном итоге продлить активный цикл клиента и увеличить общее потребление. Метод внедрения заключается в заблаговременном вмешательстве с помощью модели предупреждения об оттоке (например, предлагая купоны «скидка 15 % при покупке свыше 50» клиентам, которые не совершали покупки в течение 30 дней), а также в стимулировании преимуществ уровня членства (члены Gold получают эксклюзивные скидки на продукты).

Сокращение затрат на привлечение клиентов. Стратегия заключается в том, чтобы рекомендовать новых пользователей старым членам. Обе стороны получают баллы (1 новый пользователь = 200 баллов, которые можно обменять на 10 йен) и сопоставляют рекламные группы на основе тегов клиентов (например, матери, реклама товаров для матерей и детей).

Поддерживает визуальные условия обновления конфигурации, а изменение правил не требует перезапуска службы. Каждое утро пакетные задачи обработки используются для сканирования пользовательских данных, автоматического запуска обновлений и понижений, а также уведомления пользователей.

Решение для обеспечения согласованности в конечном итоге: когда пользователи зарабатывают баллы за потребление, сначала вычитается пул баллов, а после успешной оплаты заказа подтверждается зачисление баллов. В случае неудачной оплаты система автоматически откатит удержанные баллы.

Создавайте конкурентные барьеры для повышения лояльности участников, устанавливайте эксклюзивные преимущества для участников (например, более низкие пороги для бесплатной доставки, эксклюзивные права на предварительную продажу продукции) для увеличения расходов на переключение клиентов, корректируйте покупки на основе данных о предпочтениях клиентов, а клиенты могут использовать баллы супермаркетов для погашения членства на видеоплатформе, купонов на кофе и т. д. Сотрудничайте с местными спортзалами, например, предоставляйте участникам еженедельный абонемент при тратах свыше 300 иен, чтобы добиться двустороннего движения.

Высокая степень параллелизма и стабильности конструкции с использованием многоуровневого кэша, первый уровень – локальный кэш (например, Caffeine) для хранения горячих данных (например, базовой информации об участниках), действительный в течение 30 секунд. Второй уровень – это метка кэша кластера Redis и конфигурация активности, которая повышает доступность за счет репликации «главный-подчиненный» и разделения чтения и записи. Добавьте распределенные блокировки к пустым значениям или горячим клавишам, позволяя только одному потоку запрашивать базу данных обратно к источнику, предотвращая проникновение большого количества запросов в базу данных. В пиковый период маркетинговой активности частота запросов интерфейса QPS автоматически ограничивается (например, 5000 раз в секунду), а для запросов, превышающих лимит, возвращается результат по умолчанию (например, «активность слишком популярна, повторите попытку позже»). Когда служба тегов недоступна, она автоматически переключается на резервное правило. В этой системе имеется 11 таблиц данных. В данной статье основное внимание уделяется 3 из них: `t_user`, `t_goods`, `t_supplier`, `t_supplier`, `t_user`.

Таблица администратора в основном используется для хранения информации о менеджерах продуктовых магазинов, такой как номер учетной записи, пароль, идентификатор, псевдоним, разрешения и другая информация администратора. Сущность администраторов показана в таблице 5 ниже.

Таблица 5 – Страница администратора

Серийный номер	Имя	Тип	Длина	Иллюстрировать
1	2	3	4	5
1	id	int	11	Основной ключ
2	bz	varchar	100	Имя примечания
3	password	varchar	255	Пароль
4	true_name	varchar	50	Имя
5	user_name	varchar	50	Имя пользователя
6	remarks	varchar	1000	Примечания

t_goods (Список продуктов):

Таблица продуктов в основном используется для хранения информации о продукте, такой как код продукта, модель продукта, название продукта, производитель, цена покупки, цена продажи и другая информация. Список продуктов показан в Таблице 6 ниже.

Таблица 6 – Список продуктов

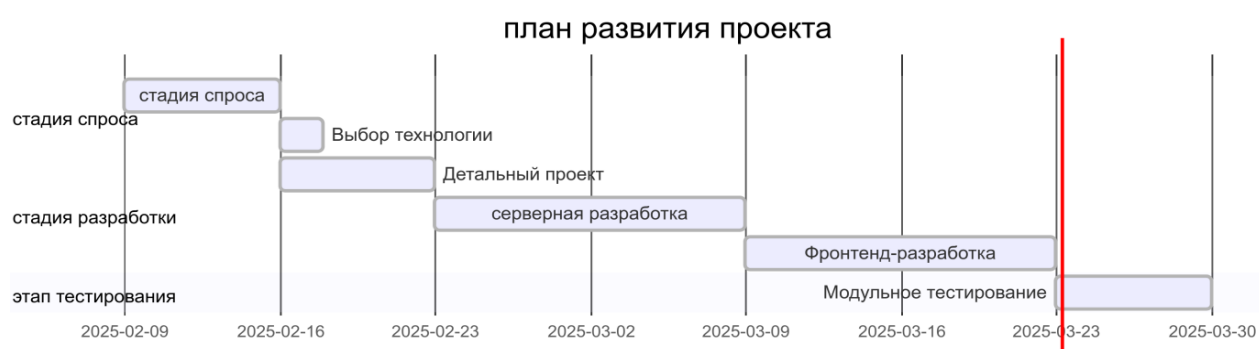
Серийный номер	Имя	Тип	Длина	Иллюстрировать
1	2	3	4	5
1	id	int	11	Основной ключ
2	code	varchar	50	Код продукта
3	inventory_quantity	int	11	Количество на складе
4	min_num	int	11	Минимальный запас
5	model	varchar	50	Модель продукта
6	name	varchar	50	наименование товара
7	producer	varchar	200	производитель
8	purchasing_price	float	0	Цена
9	remarks	varchar	1000	Примечание
10	selling_price	float	0	Цена продажи
11	unit	varchar	10	Товарная единица
12	type_id	int	11	Категория продукта
13	state	int	11	Статус продукта
14	last_purchasing_price	float	0	Последний раз

t_supplier (Таблица поставщиков). В таблице поставщиков в основном хранится информация о поставщике, такая как имя поставщика, контактная информация, адрес и другая информация. Таблица поставщиков показана в таблице 7 ниже.

Таблица 7 – Таблица поставщиков

Серийный номер	Имя	Тип	Длина	Иллюстрировать
1	id	int	11	основной ключ
2	address	varchar	300	Контактный адрес
3	contact	varchar	50	Контактное
4	name	varchar	200	Наименование поставщика
5	number	varchar	50	Контактный номер
6	remarks	varchar	1000	Примечание

В процессе разработки системы, чтобы уточнить планирование времени и распределение ресурсов для задач на каждом этапе, диаграмма Ганта является основным инструментом для визуального отображения хода проекта. Он может четко отображать время начала и завершения задачи, параллельные/последовательные связи и этапы, помогая команде координировать ритм разработки и избегать рисков задержки, как показано на рисунке 24.



Рисунке 24 – Диаграмма Ганта

Локальное развертывание системы управления продуктовым магазином:

сочетая техническую структуру Spring Boot, базу данных MySQL и интегрированную среду разработки IntelliJ IDEA, он охватывает создание среды, настройку базы данных, построение системы, тестовую проверку и т. д. с целью предоставить низкопороговое, недорогое и высоконадежное решение для развертывания для сценариев мелкой розничной торговли с ограниченными ресурсами.

Выбор технологии и анализ преимуществ: Spring Boot имеет встроенный сервер Tomcat для упрощения процесса развертывания и позволяет быстро интегрировать функциональные модули через зависимости Starter;

MySQL поддерживает управление транзакциями, чтобы обеспечить атомарность кассовых операций и инвентаризации; версия для сообщества бесплатна, что снижает затраты на лицензирование.

IntelliJ IDEA предоставляет интеллектуальные подсказки по коду и управление зависимостями Maven для повышения эффективности разработки.

Уровень пользовательского интерфейса (уровень представления) обеспечивает интерфейс взаимодействия с пользователем и поддерживает доступ к нескольким терминалам (кассир, фон управления, мобильный терминал).

Техническая реализация: настольное приложение на базе JavaFX или Swing, адаптированное к периферийным устройствам, таким как сканеры кода и кассовые боксы. Реализуйте веб-интерфейс с помощью Thymeleaf или Vue.js.

Уровень бизнес-логики (основной уровень). Управление товарами: обеспечивает ведение информации о продуктах, управление классификацией и корректировку цен.

Обработка заказов: реализация кассовой логики, вычет запасов в реальном времени и процесс возврата.

Управление участниками: правила накопления баллов, уровни членства и стратегии скидок.

Анализ отчетов: создавайте ежедневные/ежемесячные отчеты, поддерживайте рейтинги популярных продуктов и предупреждайте о медленно продаваемых продуктах.

Техническая реализация: сервис Spring Boot, инкапсулирующий бизнес-логику с помощью аннотации `@Service`.

Уровень доступа к данным (уровень персистентности) инкапсулирует операции с базой данных и предоставляет унифицированный интерфейс доступа к данным. Техническая реализация: используйте Spring Data JPA или MyBatis для работы с MySQL. Обеспечьте согласованность данных с помощью управления транзакциями (`@Transactional`).

Уровень инфраструктуры (уровень поддержки) регулярно каждый день экспортирует файлы SQL в локальное или облачное хранилище. Logback записывает журналы операций для облегчения аудита и устранения неполадок. Реализуйте контроль доступа к ролям на основе Spring Security. Уровень внешнего обслуживания (уровень интеграции) взаимодействует с платежным API Alipay/WeChat и обрабатывает обратные вызовы транзакций. Управляйте термопринтером с помощью набора команд ESC/POS.

Базовая конструкция среды. Установка JDK: выберите JDK 17 и выше, чтобы соответствовать эксплуатационным требованиям Spring Boot 3.x. Установка MySQL: разверните версию MySQL 8.0, созданную сообществом, и настройте набор символов `utf8mb4` для поддержки хранилища на китайском языке. Конфигурация IDEA: установите плагин Spring Boot и настройте образ хранилища Maven для ускорения загрузки зависимостей. Сканер: имитирует ввод с клавиатуры и фиксирует данные штрих-кода посредством мониторинга событий.

Чековый принтер: вызовите набор команд ESC/POS для получения шаблонного вывода формата чека. Управляйте Spring Boot Starter, MySQL Connector и другими зависимостями унифицированно через Maven. Запустите пакет `mvn clean`, чтобы создать исполняемый пакет JAR со встроенным сервером Tomcat.

3.2. Примеры фактического тестирования программного продукта

План испытаний направлен на проверку функциональной целостности,

стабильности работы и безопасности системы с помощью систематических методов тестирования, чтобы гарантировать, что она соответствует реальным эксплуатационным потребностям продуктового магазина. Тест охватывает четыре основных модуля: функциональное тестирование, тестирование производительности, тестирование безопасности и тестирование совместимости, а также сочетает в себе методы тестирования «черного» и «белого ящика» для создания плана тестирования, который можно реализовать.

Функциональное тестирование для проверки нормальности основных процессов, таких как кассовый сбор, управление товарами и синхронизация запасов, с использованием тестирования черного ящика (основное) + тестирования белого ящика (вспомогательное)

Тест производительности, чтобы убедиться, что система поддерживает одновременную проверку нескольких терминалов (не менее 3 устройств работают одновременно), время отклика ≤ 2 секунды, используется тестирование черного ящика

Тестирование безопасности для предотвращения SQL-инъекций, несанкционированного доступа и других уязвимостей для обеспечения конфиденциальности данных с использованием тестирования белого ящика + инструментального сканирования.

Тестирование совместимости, адаптация к различным операционным системам (Windows 10/11), браузерам (Chrome/Edge) и аппаратным устройствам (сканерам, моделям принтеров), с использованием тестирования методом черного ящика. Разработка функционального тестового примера. Содержание теста показано в таблице 8 и таблице 9.

Таблица 8 – Тестирование процесса оформления заказа

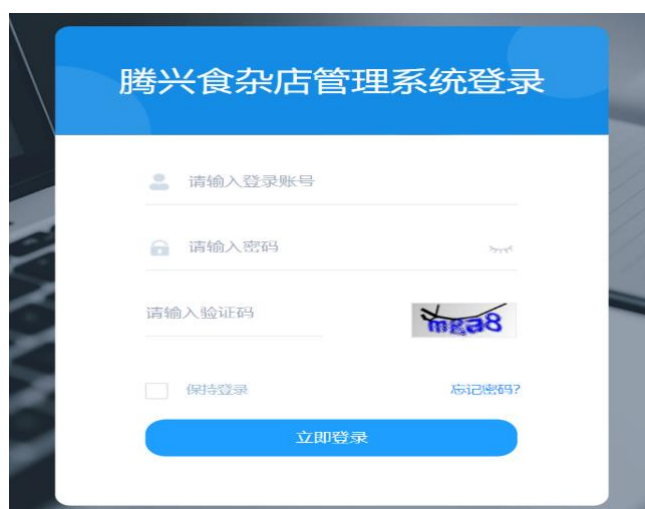
Тестовые задания	Ввод/Операция	Ожидаемые результаты
1	2	3
Обычный кассир	Сканирование продукта А (достаточный запас) → платеж успешен	Формирование заказов, учет запасов.

Недостаточная проверка запасов	Сканировать предмет В (инвентарь равен 0)	Распечатайте чек с указанием названия продукта, количества и общей стоимости.
Продолжение таблицы 8		
Комбинированная оплата	Оплатите часть суммы наличными + отсканируйте QR-код, чтобы оплатить оставшуюся сумму.	Накопление членских баллов (если вы войдете в свою учетную запись участника)

Таблица 9 – Тест управления товарами

Тестовые задания	Ввод/Операция	Ожидаемые результаты
1	2	3
Добавить новый продукт	Введите название, цену, штрих-код → нажмите «Сохранить».	В списке продуктов отображаются новые элементы, а записи базы данных соответствуют входным данным.
Изменение цены	Скорректируйте цену продукта А с 10 до 12 юаней.	Цена за единицу обновлена, историческая цена не изменилась
Пакетный импорт	Загрузите CSV-файл (содержащий 100 элементов данных о продукте)	Система выдаст сообщение «Импорт выполнен успешно», и в список товаров будет добавлено 100 записей.

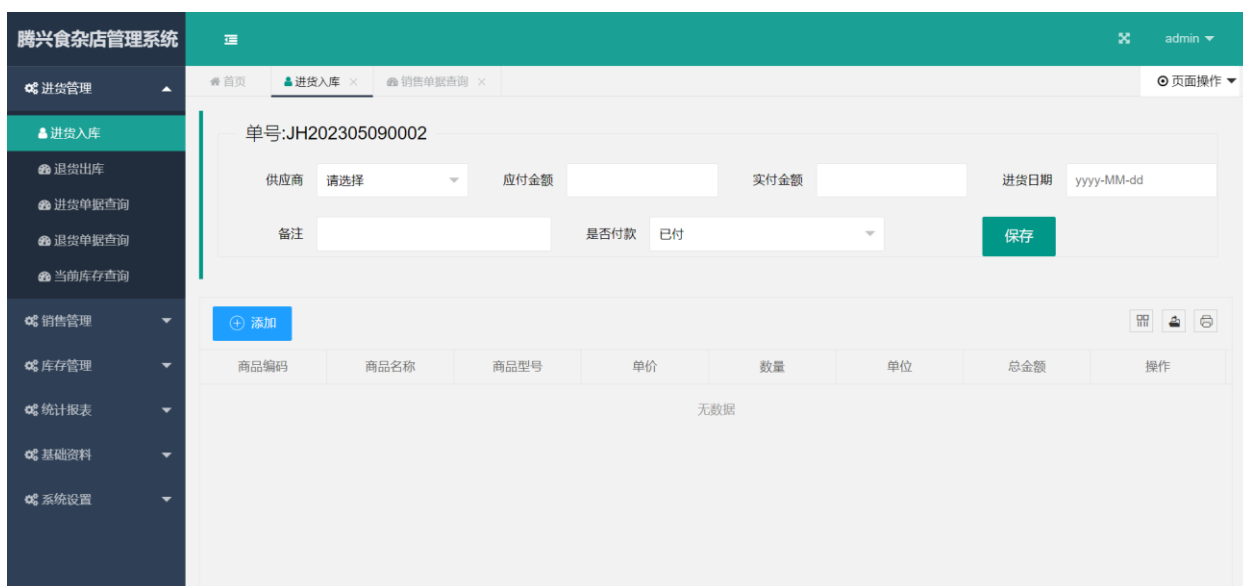
Интерфейс входа в систему. это портал ко всей системе. После ввода правильного имени пользователя и пароля система автоматически проверит базу данных. Только после успешного сопоставления данных в базе данных вы сможете войти на домашнюю страницу системы. Интерфейс входа в систему управления показан на рисунке 25.



Рисунке 25 – Тест входа

После успешного входа в систему вы сначала попадаете на домашнюю страницу системы. Домашняя страница содержит приветственное сообщение, функциональные модули слева и некоторые функции выше. Администраторы могут выбирать соответствующие модули на главной странице для работы. Интерфейс приветственного сообщения на Домашняя страница создана с использованием компонента карусели «Карусель». Слева подробные функции созданы с использованием форм.

Возьмем в качестве примера управление закупками, как показано на рисунке 26.



Рисунке 26 – Управление закупками

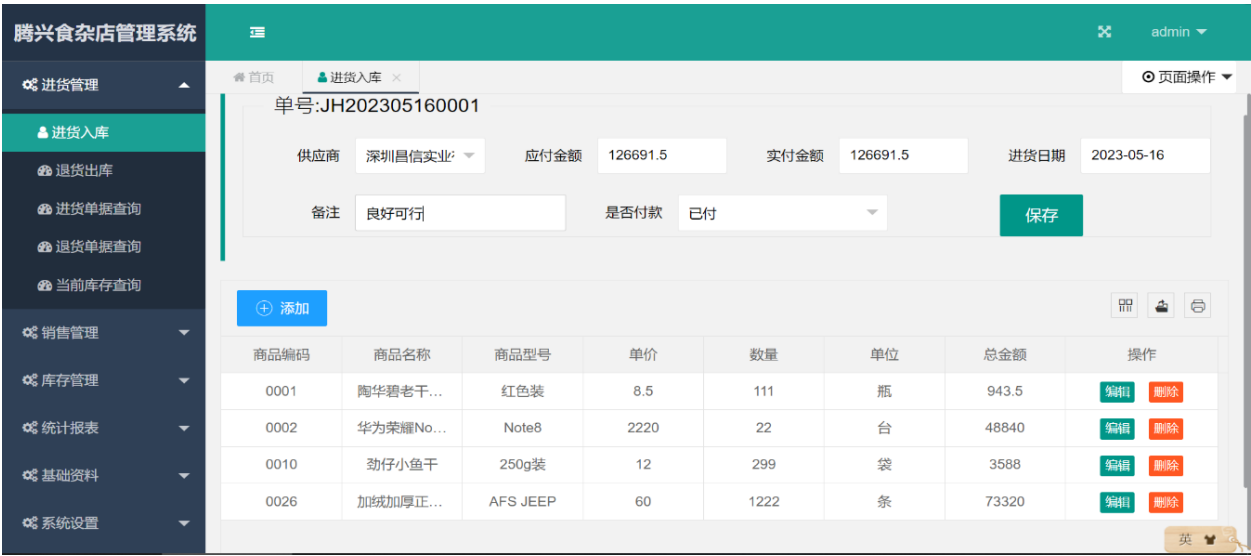
Модуль управления закупками представляет собой ключевой компонент

информационной системы, предназначенный для автоматизации и оптимизации процессов, связанных с приобретением товарно-материальных ценностей. Он включает в себя широкий спектр функциональных возможностей, таких как регистрация и учёт закупок, оформление заявок на закупку, обработка возвратов поставщикам, а также формирование заявок на возврат продукции.

С помощью данного модуля менеджеры розничных торговых точек получают возможность централизованного управления всей информацией, связанной с входящими (поступающими от поставщиков) и исходящими (возвращаемыми поставщикам) товарами. Это обеспечивает прозрачность процессов, контроль за соблюдением договорных условий и своевременное обновление товарных остатков.

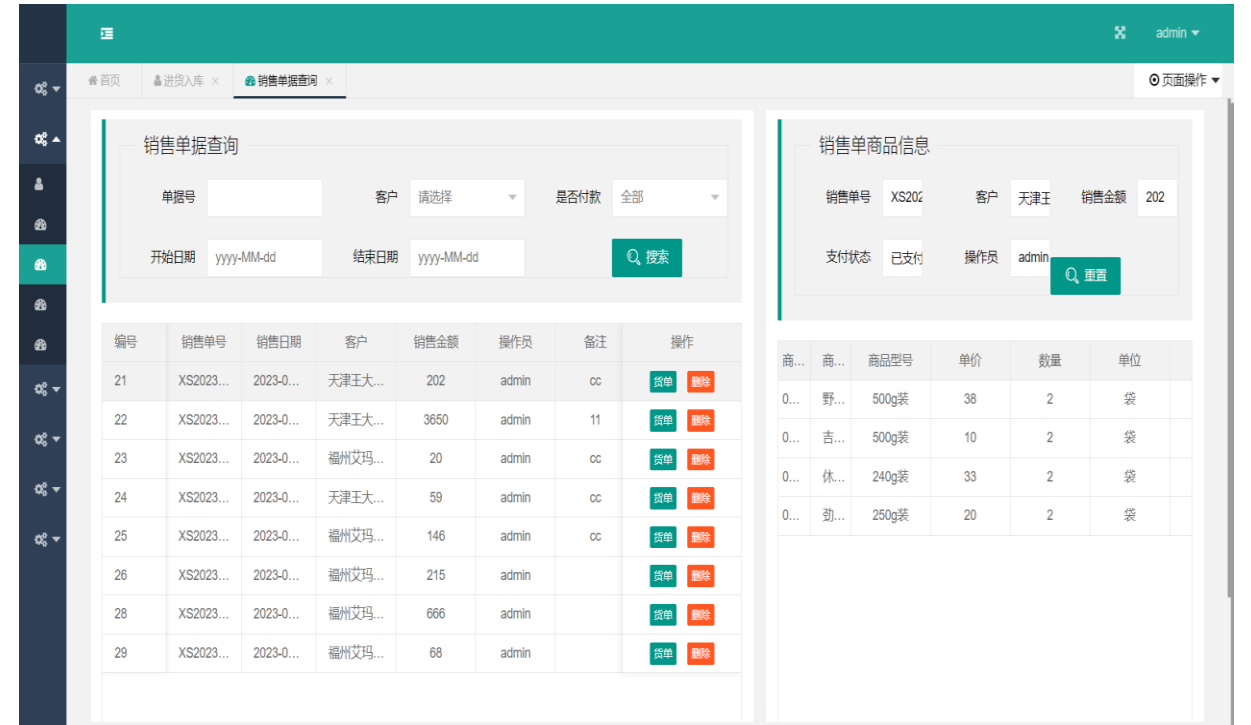
Процесс регистрации закупки предполагает внесение информации о поставщике, номере заказа, дате закупки, общей сумме к оплате, фактически уплаченной сумме, а также других сопровождающих данных (например, номера накладной, условия поставки, ответственного лица и прочее). После сохранения этих данных система отображает уведомление об успешной регистрации операции и предоставляет доступ к соответствующим историческим записям.

Исторические данные формируются автоматически и включают в себя детализированную информацию обо всех совершённых закупках и возвратах, что позволяет осуществлять анализ динамики закупочной деятельности, контролировать бюджеты, выявлять несоответствия, а также оценивать эффективность работы с поставщиками. Структура и содержание отображаемых сведений представлены на рисунке 27, где продемонстрирован пример пользовательского интерфейса со сводкой успешно завершённой закупочной операции.



Рисунке 27 – Тест модуля управления закупками

Модуль управления продажами включает в себя такие функции, как отгрузка проданного товара, возвраты товаров клиентами, запросы на торговые документы и запросы на возврат товаров клиентами. В этом тесте в качестве примера взято приложение для обработки заказов на продажу. В форму вводятся ключевые слова, определяется текст для нечеткого поиска, а также задаются временные атрибуты. После ввода данных они извлекаются из базы данных и отображаются в правом интерфейсе. Результаты испытаний показаны на рисунке 28.



Рисунке 28 – Тест по управлению продажами

Решение для тестирования производительности использует инструмент тестирования JMeter.

Сценарий предназначен для однопользовательской кассы: смоделируйте работу кассового терминала и измерьте среднее время отклика (ожидаемое ≤ 1 секунды). Параллельная работа с несколькими пользователями: смоделируйте кассовую операцию на трех терминалах одновременно в течение 10 минут, чтобы проверить стабильность системы и использование ресурсов (ЦП $< 80 \%$, без переполнения памяти).

С помощью этого плана тестирования можно полностью проверить функциональность, надежность и простоту использования «Системы управления продуктовым магазином», чтобы гарантировать ее стабильную работу в сценариях мелкой розничной торговли.

ЗАКЛЮЧЕНИЕ

В ходе разработки системы управления продуктовым магазином были консолидированы полученные в процессе обучения знания и навыки, которые нашли практическое применение при проектировании архитектуры и реализации основных функциональных модулей. Проектная деятельность велась в соответствии с ключевыми этапами жизненного цикла программного обеспечения, что обеспечило логичную структуру, модульность, возможность масштабирования и адаптации системы под реальные потребности небольших розничных торговых точек.

Созданная система отвечает характерным требованиям продуктового магазина, включая учёт закупок и продаж, управление остатками, автоматизацию кассовых операций и генерацию отчётности. Основу программной реализации составляет стек технологий Spring Boot и MySQL, что обеспечивает высокую доступность, гибкость и совместимость с существующей ИТ-инфраструктурой. Использование кэша Redis и транзакционного управления JPA позволило достичь надёжной консистентности данных в условиях многопользовательской нагрузки.

Система реализована в концепции «лёгкий код – тяжёлый процесс», где упор сделан на минималистичный, интуитивно понятный интерфейс при сохранении глубокой бизнес-логики на уровне серверной части. Это снижает требования к обучению персонала и повышает общее удобство использования. Предложенное решение предназначено для локализованного развертывания на одном компьютере или с возможностью одновременной работы нескольких терминалов, что делает его особенно актуальным для магазинов с ограниченным бюджетом на цифровую трансформацию. Общая стоимость внедрения составляет менее 20 % от стоимости аналогичных коммерческих решений, при этом отсутствует необходимость в регулярных лицензионных платежах.

Тем не менее, в процессе разработки были выявлены и определённые

ограничения. В частности, из-за ограниченного времени и опыта проектирование и реализация некоторых компонентов не достигли необходимой глубины. Система требует дальнейшего тестирования, повышения производительности, улучшения пользовательского опыта и обеспечения безопасности на уровне доступа и хранения данных. В текущем виде система реализует базовую функциональность, но закладывает прочный фундамент для её расширения и масштабирования.

В перспективе планируется дальнейшее развитие проекта: внедрение облачных технологий для удалённого доступа и синхронизации между торговыми точками, интеграция с мобильными приложениями, использование методов аналитики и машинного обучения для прогнозирования спроса и оптимизации товарных остатков. Также возможна адаптация системы под другие сегменты малого бизнеса.

В целом, выполненная работа продемонстрировала способность применять теоретические знания на практике, выявлять проблемы в реальной предметной области и предлагать эффективные программные решения. Полученный опыт будет полезен как для будущей профессиональной деятельности, так и для продолжения научных исследований в области разработки прикладного программного обеспечения для розничной торговли.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Аветисян, Т. В. Учебно-исследовательская система управления персоналом в киберфизической системе [Электронный ресурс] / Т. В. Аветисян, К. И. Львович, Э. М. Львович // ИАС. – 2023. – №3. – Режим доступа : <https://cyberleninka.ru/article/n/uchebno-issledovatel'skaya-sistema-upravleniya-personalom-v-kiberfizicheskoy-sisteme>. – 15.05.2025.

2 Андреасян, Г. М. Системы управления информацией [Электронный ресурс] / Г. М. Андреасян, М. Е. Федоров // Экономика и социум. – 2021. – №4-1. – С. 83. – Режим доступа : <https://cyberleninka.ru/article/n/sistemy-upravleniya-informatsiey>. – 15.05.2025.

3 Байрамов, Р. Д. Система управления студенческим общежитием [Электронный ресурс] / Р. Д. Байрамов, Ш. Чарыев // Символ науки. – 2024. – №12-2-1. – Режим доступа : <https://cyberleninka.ru/article/n/sistema-upravleniya-studencheskim-obschezhitiey>. – 15.05.2025.

4 Балашова, В. В. Автоматизированная система управления предприятием в системе менеджмента качества: управление производством [Электронный ресурс] / В. В. Балашова, Д. П. Ветров, И. А. Картушин, Г. Н. Карпенко // НиКа. – 2005. – Режим доступа : <https://cyberleninka.ru/article/n/avtomatizirovannaya-sistema-upravleniyapredpriyatiem-v-sisteme-menedzhmenta-kachestva-upravlenie-proizvodstvom>. – 15.05.2025.

5 Безрученко, В. А. Управление товарными запасами. Реализация эффективной системы управления товарными запасами на складах / В. А. Безрученко. – Самара : Полиграфическое объединение «Стандарт». – 2020. – 530 с.

6 Брусакова, И. А. Система управления базами измерительных знаний [Электронный ресурс] / И. А. Брусакова, С. О. Мамаева // Прикладная информатика. – 2006. – №5. – Режим доступа : <https://cyberleninka.ru/article/n/sistema-upravleniya-bazami-izmeritelnyh-znaniy>. – 15.05.2025.

7 Винокуров, А. А. Современные технологии ревеню-менеджмента в действии в розничной торговле [Электронный ресурс] / А. А. Винокуров //

Экономика и бизнес: теория и практика. – 2020. – №5-2. – Режим доступа : <https://cyberleninka.ru/article/n/sovremennye-tehnologii-revenyu-menedzhmenta-v-deystvii-v-roznichnoy-torgovle>. – 15.05.2025.

8 Грибова, В. В. Процессы управления интеллектуальными системами [Электронный ресурс] / В. В. Грибова, А. С. Клещев // Онтология проектирования. – 2011. – №1. – С. 2. – Режим доступа : <https://cyberleninka.ru/article/n/protsessy-upravleniya-intellektualnymi-sistemami>. – 15.05.2025.

9 Шереметьева, Е. Н. Система управления цифровыми талантами [Электронный ресурс] / Е. Н. Шереметьева, А. К. Фролова, Е. П. Баринова // Вестник АГТУ. Серия: Экономика. – 2024. – №2. – Режим доступа : <https://cyberleninka.ru/article/n/sistema-upravleniya-tsifrovymi-talantami>. – 15.05.2025.

10 Карибаев, А. А. Место и роль системы управления знаниями в единой системе управления предприятием [Электронный ресурс] / А. А. Карибаев, А. А. Камиева // In The World Of Science and Education. – 2024. – №20. – Режим доступа : <https://cyberleninka.ru/article/n/mesto-i-rol-sistemy-upravleniya-znaniyami-v-edinoy-sisteme-upravleniya-predpriyatiem>. – 15.05.2025.

11 Козлов, К. И. Экспертные системы в управлении IT-проектами [Электронный ресурс] / К. И. Козлов // Вестник ХГУ им. Н. Ф. Катанова. – 2024. – №4. – С. 50. – Режим доступа : <https://cyberleninka.ru/article/n/ekspertnye-sistemy-v-upravlenii-it-proektami>. – 15.05.2025.

12 Козлов, К. И. Экспертные системы в управлении it-проектами // Вестник ХГУ им. Н. Ф. Катанова. 2024. №4. – С. 50. – Режим доступа : <https://cyberleninka.ru/article/n/ekspertnye-sistemy-v-upravlenii-it-proektami>

13 Кравченко, К. А. Применение нейросетевых технологий в прогнозировании продаж / К. А. Кравченко, В. В. Ткаченко // В сборнике: Наука XXI века: проблемы, перспективы и актуальные вопросы развития общества, образования и науки. Материалы международной межвузовской осенней научно-практической конференции. – 2020. – С. 145-150.

14 Леонтьева, В. М. Пути совершенствования управления товароснабжением универсамов: автореф. дис. кандидата эконом. наук : 08.00.05 / В. М. Леонтьева. – Москва : Заочный институт советской торговли. – 1981. – 25 с.

15 Логачева, Н. В. Важность тестирования программного обеспечения в процессе разработки программного обеспечения [Электронный ресурс] / Н. В. Логачева, М. Л. Ладоничева, К. С. Пузырева // Инновационная наука. – 2022. – №2-2. – Режим доступа : <https://cyberleninka.ru/article/n/vazhnost-testirovaniya-programmnogo-obespecheniya-v-protssesse-razrabotki-programmnogoobespecheniya>. – 15.05.2025.

16 Макаров, Г. В. Развитие методов и алгоритмов теории подобия для систем управления: дис. кандидата технических наук : 2.3.3 / Г. В. Макаров // ФГБОУ ВО «Томский государственный университет систем управления и радиоэлектроники». – Новокузнецк. – 2021. – 135 с.

17 Малинецкий, Г. Г. Системы управления наукой России [Электронный ресурс] / Г. Г. Малинецкий // SAEC. – 2023. – №1. – Режим доступа : <https://cyberleninka.ru/article/n/sistemy-upravleniya-naukoy-rossii>. – 15.05.2025.

18 Маммедова, Г. Цифровизация общества: система управления изменениями [Электронный ресурс] / Г. Маммедова // Вестник науки. – 2023. – №6 (63). – Режим доступа : <https://cyberleninka.ru/article/n/tsifrovizatsiya-obshchestva-sistema-upravleniya-izmeneniyami>. – 15.05.2025.

19 Меняев, М. Ф. Цифровизация в системе управления качеством предприятия [Электронный ресурс] / М. Ф. Меняев // Россия: тенденции и перспективы развития. – 2023. – №18-1. – Режим доступа : <https://cyberleninka.ru/article/n/tsifrovizatsiya-v-sisteme-upravleniya-kachestvom-predpriyatiya>. – 15.05.2025.

20 Минтуш, О. В. Динамическая система управления организациями [Электронный ресурс] / О. В. Минтуш // Инновации и инвестиции. – 2023. – №4. – Режим доступа : <https://cyberleninka.ru/article/n/dinamicheskaya-sistema-upravleniya-organizatsiyami>. – 15.05.2025.

21 Митришкин, Ю. В. Синтез систем управления: учебное пособие по

курсу «Математические методы разработки исследования систем управления» / Ю. В. Митришкин. – Москва : Физический факультет МГУ. – 2023. – 300 с.

22 Мурадова, А. О. Система управления библиотекой [Электронный ресурс] / А. О. Мурадова // ELS. – 2024. – №2. – Режим доступа : <https://cyberleninka.ru/article/n/sistema-upravleniya-bibliotekoy>. – 15.05.2025.

23 Направления автоматизации системы управления вузом [Электронный ресурс]. – Режим доступа : <https://cyberleninka.ru/article/n/napravleniya-avtomatizatsii-sistemy-upravleniya-vuzom/viewer>. – 15.05.2025.

24 Несговорова, В. С. Построение систем управления процессно-ориентированных организаций: дис. кандидата экономических наук : 08.00.05 / В. С. Несговорова // ФГБОУ ВО «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)». – Москва. – 2022. – 161 с.

25 Панасенко, С. В. Направления развития современной розничной торговли [Электронный ресурс] / С. В. Панасенко, И. Б. Стукалова, Т. А. Мазунина // Российское предпринимательство. – 2018. – №3. – Режим доступа : <https://cyberleninka.ru/article/n/napravleniya-razvitiya-sovremennoy-roznichnoy-torgovli>. – 15.05.2025.

26 Сапунов, А.В. Переход к цифровой системе управления в организациях [Электронный ресурс] / А. В. Сапунов // ЕГИ. – 2023. – №1 (45). – Режим доступа : <https://cyberleninka.ru/article/n/perehod-k-tsifrovoy-sisteme-upravleniya-v-organizatsiyah>. – 15.05.2025.

27 Соловьёв, А. В. Основы автоматики и теория управления техническими системами: курс лекций для студентов, обучающихся по направлению подготовки 26.05.06 «Эксплуатация судовых энергетических установок» / А. В. Соловьёв, Е. Н. Поселенов // Федеральное агентство морского и речного транспорта, Федеральное государственное бюджетное образовательное учреждение высшего образования «Волжский государственный университет водного транспорта», Кафедра систем информационной безопасности, управления и телекоммуникаций. – Нижний Новгород : ВГУВТ. – 2022. – 104 с.

28 Стержанов, М. В. Разработка индекатора системы управления документами [Электронный ресурс] / М. В. Стержанов // Доклады БГУИР. – 2012. – №6. – С. 68. – Режим доступа : <https://cyberleninka.ru/article/n/razrabotka-indeksatora-sistemy-upravleniya-dokumentami>. – 15.05.2025.

29 Ткаченко, В. В. Информационная система поддержки онлайн-продаж продуктовых супермаркетов с использованием современных алгоритмов и коммуникационных технологий [Электронный ресурс] / В. В. Ткаченко, Е. О. Волков, Е. М. Силин. – 2022. – №6. – С. 44. – Режим доступа : <https://cyberleninka.ru/article/n/informatsionnaya-sistema-podderzhki-onlayn-prodazh-produktovyh-supermarketov-s-ispolzovaniem-sovremennyh-algoritmov-i>. – 15.05.2025.

30 Харева, В. А. Современные тенденции развития электронной торговли [Электронный ресурс] / В. А. Харева, Д. А. Жаркова // Научный вестник Ю-ИМ. – 2019. – №4. – Режим доступа : <https://cyberleninka.ru/article/n/sovremennye-tendentsii-razvitiya-elektronnoy-torgovli>. – 15.05.2025.

31 Чжун, Л. Разработка проектов Spring Boot на уровне предприятия от начального уровня до мастерства ПЭВМ : учебное пособие / Л. Чжун, Ц. Ло // Хубэй : Хуачжунский университет науки и технологий. – 2020. – С. 53–64.