

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра дифференциальных уравнений и математической экономики

**Разработка системы автоматизации продаж и клиентского
сервиса для кондитерского бизнеса на основе чат-бота**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 451 группы

направления **38.03.05 – Бизнес-информатика**

механико-математического факультета

Ручко Анастасии Дмитриевны

Научный руководитель
зав.кафедрой, д.ф.-м.н.,
доцент

В.С.Рыхлов

Заведующий кафедрой
зав.кафедрой, д.ф.-м.н.,
доцент

В.С.Рыхлов

Саратов 2026

Введение. Объём российского рынка кондитерской продукции в 2025 году превысил 2 трлн рублей, при этом всё большую роль играют онлайн-продажи. Для малых и средних кондитерских предприятий это создаёт дополнительные сложности: крупные компании обладают развитой цифровой инфраструктурой и могут обеспечивать более высокий уровень автоматизации и клиентского сервиса. В связи с этим возрастает потребность в доступных решениях, позволяющих небольшим предприятиям оптимизировать работу с клиентами и повысить эффективность бизнес-процессов.

Актуальность данной бакалаврской работы связана с высокой нагрузкой на сотрудников при обработке клиентских обращений. Консультирование по ассортименту, ответы на часто задаваемые вопросы, приём заказов, сопровождение клиентов и работа с отзывами требуют значительных временных затрат. При этом большая часть подобных операций является типовой и может быть автоматизирована без ухудшения качества обслуживания. В этих условиях необходимой становится разработка системы, сочетающей автоматизацию транзакционных операций с интеллектуальными модулями обработки естественного языка.

Целью бакалаврской работы является разработка прототипа чат-бота, направленного на автоматизацию процесса продаж в сфере кондитерского бизнеса, с последующим анализом экономической эффективности данного решения

Для достижения поставленной цели были решены следующие задачи:

- провести анализ российского рынка кондитерских изделий и существующих решений в области автоматизации клиентского сервиса;
- выполнить моделирование бизнес-процессов в нотациях «AS-IS» и «TO-BE», а также определить функциональные и нефункциональные требования к системе;
- спроектировать архитектуру системы и обосновать выбор используемых технологий;
- реализовать клиентский и административный модули системы на основе требований;
- провести тестирование системы, выполнить оценку экономической эффективности и анализ возможных рисков внедрения.

Основное содержание работы. Работа состоит из введения, трёх разделов, заключения, списка использованных источников и приложений.

Работа включает следующие три раздела:

1. Анализ предметной области
2. Разработка системы автоматизации на основе чат-бота
3. Оценка эффективности и бизнес-результатов

В первом разделе проведён комплексный анализ предметной области: состояние рынка, конкурентные предложения, моделирование бизнес-процессов и формулирование требований к системе.

Анализ рынка кондитерских изделий показал, что в 2025 году объём продаж в денежном выражении вырос на 16% и достиг 2,16 трлн рублей. Доля отечественной продукции составила 93%, а отпускные цены на торты увеличились на 13,5%. Среди значимых потребительских трендов выделяются нарастающий запрос на персонализацию ассортимента, рост спроса на функциональные продукты без сахара и глютена, а также выраженная сезонность, предъявляющая повышенные требования к пропускной способности каналов приёма заказов.

С период 2020 по 2025 год доля малого бизнеса в сфере общественного питания, использующего цифровые решения, выросла с 35 до 60%. Перспективным направлением стала «торговля через диалог» — оформление заказов и консультации в мессенджерах без перехода на сайт или приложение. Для небольших кондитерских такой подход привлекателен низкими затратами, популярностью мессенджеров и привычным для клиентов форматом общения, что делает чат-боты эффективным инструментом автоматизации продаж и сервиса.

Проведённый конкурентный анализ показал, что существующие решения автоматизируют лишь отдельные функции: Candy Store Bot реализует базовый интернет-магазин, Neurobot ориентирован на интеллектуальные ответы, а платформы Okibot, Ameton и Tomogu предлагают готовые, но слабо адаптируемые инструменты. Ни одно из них не объединяет в одной системе каталог, оформление заказа, интеллектуальные ответы на основе базы знаний, повторные заказы, анализ отзывов и административный интерфейс. При этом для кондитерской отрасли критически важны поддержка кастомизации

(начинка, размер, оформление) и автоматические ответы о составе, аллергиях, сроках хранения и доставке.

Моделирование бизнес-процессов выполнено в нотации BPMN 2.0, поскольку она является общепринятым стандартом бизнес-анализа и позволяет наглядно отразить последовательность операций, условия переходов и распределение ответственности между участниками процесса

Текущий процесс обработки заказов в небольшой кондитерской в основном выполняется вручную. Клиенты направляют заявки через мессенджеры, социальные сети или по телефону, после чего менеджер уточняет детали заказа, рассчитывает стоимость, согласовывает условия доставки или самовывоза, принимает подтверждение оплаты и передаёт информацию кондитеру. После изготовления продукции сотрудник дополнительно уведомляет клиента о готовности заказа.

В целевой модели значительная часть операций автоматизируется с помощью чат-бота. BPMN-диаграмма целевого процесса представлена на рисунке 1.

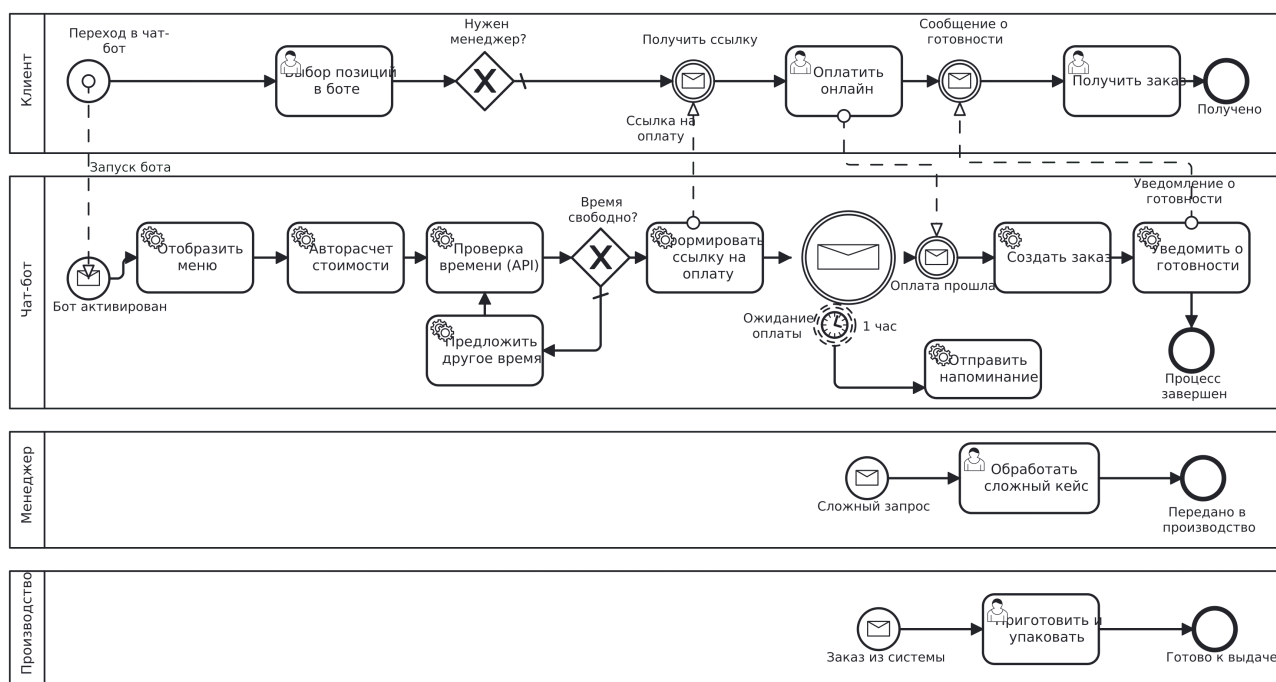


Рисунок 1 – BPMN-диаграмма процесса «ТО-ВЕ»

После внедрения системы большая часть типовых операций выполняется без участия сотрудников. В результате снижается нагрузка на персонал,

уменьшается количество ошибок, обеспечивается круглосуточный приём заказов и сокращается время обработки обращений. Сотрудники могут сосредоточиться на контроле качества продукции и решении нестандартных ситуаций, а бизнес получает возможность масштабировать продажи без пропорционального увеличения затрат.

На основании проведённого анализа сформулированы три уровня требований к системе. Бизнес-требования определяют стратегические цели: автоматизация 70% входящих обращений, сокращение времени оформления заказа, повторный заказ в один клик с ростом частоты покупок и устранение потерь ночного трафика. Функциональные требования определили основные задачи, которые должна выполнять система. Нефункциональные требования задают измеримые показатели качества: время ответа бота менее 1 секунды, время ответа NLP-модуля — не более 3 секунд, доступность не ниже 99,9% и хранение персональных данных исключительно на серверах в Российской Федерации.

Во втором разделе спроектирована архитектура системы и базы данных, а также реализованы клиентский и административный модули.

Система построена по трёхуровневой архитектуре, включающей уровень представления, уровень бизнес-логики и уровень данных. Уровень представления обеспечивает взаимодействие пользователя с системой через мессенджер Telegram. Бизнес-логика отвечает за функционал системы. Уровень данных отвечает за хранение информации о пользователях, товарах, заказах, отзывах и документах базы знаний.

Выбор технологического стека обоснован сравнительным анализом альтернатив. Система построена на Python 3.12 с фреймворком aiogram 3.x, что обеспечивает асинхронную обработку сообщений и поддержку многошаговых сценариев оформления заказа через FSM.

Ответы на вопросы реализованы через RAG-модуль на базе фреймворка LangChain и ChromaDB, где LangChain управляет логикой взаимодействия с моделью, а ChromaDB выполняет векторный поиск с использованием эмбеддингов deepvk/USER-bge-m3 и генеративной модели Qwen3-80B. Для анализа тональности отзывов применяется RuBERT, а административная панель выполнена как десктопное приложение на CustomTkinter с визуализацией

данных через matplotlib, что исключает необходимость развёртывания отдельного веб-сервера.

В качестве СУБД используется PostgreSQL с доступом через SQLAlchemy 2.0, обеспечивающую объектно-реляционное отображение.

```
1 class User(Base):
2     __tablename__ = 'users'
3     id = Column(Integer, primary_key=True)
4     telegram_id = Column(BigInteger, unique=True, nullable=False)
5     username = Column(String(255))
6     first_name = Column(String(255))
7     last_name = Column(String(255))
8     phone = Column(String(20), nullable=True)
9     address = Column(Text, nullable=True)
10    created_at = Column(DateTime, default=datetime.utcnow)
11    orders = relationship("Order", back_populates="user",
12                          cascade="all, delete-orphan")
```

Листинг 1: Модель User на SQLAlchemy

Клиентский модуль включает три подсистемы. Модуль заказа объединяет подмодули «Каталог», «Корзина» и «Мои заказы». Каталог загружает категории и товары из базы данных с учётом доступности, формируя карточки с ценой, весом, составом и аллергенами. Для разных категорий действуют ограничения по количеству. Корзина позволяет изменять состав, для тортов запускается сценарий кастомизации с выбором веса, бисквита, начинки и покрытия.

Подмодуль «Мои заказы» отображает историю с детализацией по каждому заказу, а функция повтора позволяет перенести позиции выбранного заказа в текущую корзину с проверкой доступности товаров и ограничений, после чего пользователь направляется в корзину для оформления.

Оформление заказа реализовано через FSM с последовательным заполнением способа получения, даты, времени, адреса, телефона, комментария и референса. На завершающем этапе выполняется проверка аллергенов и формирование заказа с переносом позиций из корзины, после чего администраторы получают уведомление.

```

1 @router.callback_query(F.data == "checkout")
2 async def start_checkout(callback: types.CallbackQuery, state: FSMContext):
3     await state.update_data(telegram_id=callback.from_user.id)
4     kb = InlineKeyboardBuilder()
5     kb.row(
6         InlineKeyboardButton(text="Доставка",
7                               callback_data="delivery_type_delivery"),
8         InlineKeyboardButton(text="Самовывоз",
9                               callback_data="delivery_type_pickup"))
10    await callback.message.answer(
11        "Выберите способ получения заказа:",
12        reply_markup=kb.as_markup())
13    await state.set_state(OrderStates.delivery_type)
14    await callback.answer()

```

Листинг 2: Фрагмент FSM оформления заказа: выбор типа доставки

Модуль ответов на вопросы основан на Retrieval-Augmented-Generation (RAG) и предназначен для обработки запросов на основе внутренней базы знаний кондитерской, включающей информацию об ассортименте, составе, аллергенах, доставке и оплате. Документы разбиваются на фрагменты с перекрытием 200 символов, что сохраняет контекст между соседними частями текста. Для векторизации применяется модель `deepvk/USER-bge-m3` с нормализацией эмбеддингов, позволяющей использовать косинусное сходство при поиске релевантных фрагментов.

```

1 prompt = ChatPromptTemplate.from_messages([
2     ("system", system_prompt),
3     ("human", "Контекст:\n{context}\n\nВопрос:\n{question}")]
4 rag_chain = ({ "context": retriever | format_docs,
5               "question": RunnablePassthrough() }
6             | prompt | llm | StrOutputParser())

```

Листинг 3: Формирование RAG-цепочки

Эмбеддинги сохраняются в векторной базе Chroma. При поступлении запроса ретривер извлекает четыре наиболее релевантных фрагмента, фор-

мирующих контекст для генеративной модели. RAG-цепочка, построенная на LangChain, включает извлечение документов, формирование промпта с системным сообщением, генерацию ответа и преобразование результата.

Для анализа тональности используется дообученная модель rubert-tiny2. Исходный датасет после фильтрации и очистки составил 97 991 запись. Для устранения дисбаланса классов применён `downsampling`, в результате сформирован сбалансированный набор из 24 081 записи с разбиением 80/20. Дообучение проводилось в режиме `AutoModelForSequenceClassification` с классификационным слоем на три выхода и функцией потерь кросс-энтропии:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^3 y_{i,k} \log(\hat{y}_{i,k}) \quad (1)$$

Оптимизация выполнялась алгоритмом AdamW (`learning_rate = 5 \cdot 10^{-5}`, `batch_size = 16`, `epochs = 3`). Функция `predict_sentiment(text)` токенизирует текст, выполняет прямой проход через модель и возвращает метку класса. В системе для положительных и нейтральных отзывов отправляется благодарственное сообщение, при негативной тональности формируется уведомление администратору.

Административный модуль реализован на двух уровнях. Первый, встроенный в Telegram, служит для ответов на вопросы клиентов, на которые не смог ответить RAG-модуль (оператор получает запрос через бота и отвечает вручную). Работа функции `/admin` в боте продемонстрирована в листинге 4.

```
1 def is_admin(user_id: int) -> bool:
2     return user_id in ADMIN_IDS
3 @router.message(Command("admin"))
4 async def admin_panel(message: types.Message):
5     if not is_admin(message.from_user.id):
6         await message.answer("У вас нет доступа.")
7         return
8     await message.answer("Админ-панель",
9         reply_markup=get_admin_keyboard())
```

Листинг 4: Проверка прав администратора и вход в панель

Второй уровень — десктопная панель `admin_desktop.py` на базе библиотеки `tkinter` — предоставляет полный CRUD-цикл управления каталогом, загрузку документов для RAG-базы знаний, фильтрацию и экспорт заказов, а также просмотр аналитики продаж. Разделение уровней позволяет сочетать оперативное реагирование с мобильного устройства и детальную аналитическую работу на рабочем месте.

В третьем разделе представлены результаты тестирования системы, расчёт экономической эффективности внедрения и анализ рисков.

В ходе тестирования клиентской части чат-бота были проверены требования, сформулированные в первом разделе. Тестирование подтвердило корректную реализацию всех заявленных функций: от начала взаимодействия с ботом до оформления заказа и ответов на вопросы. Пример взаимодействия клиента с чат-ботом представлен на рисунке 2.



Рисунок 2 – Интерфейс клиентской части чат-бота

Административная панель предназначена для управления заказами, каталогом, отзывами, документами RAG и аналитикой. В ходе тестирования проверялись фильтрация заказов, изменение статусов с автоматическим уве-

домлением клиентов, операции CRUD, перестроение векторной базы Chroma, а также корректность отображения аналитических показателей. Административная панель представлена на рисунке 3.

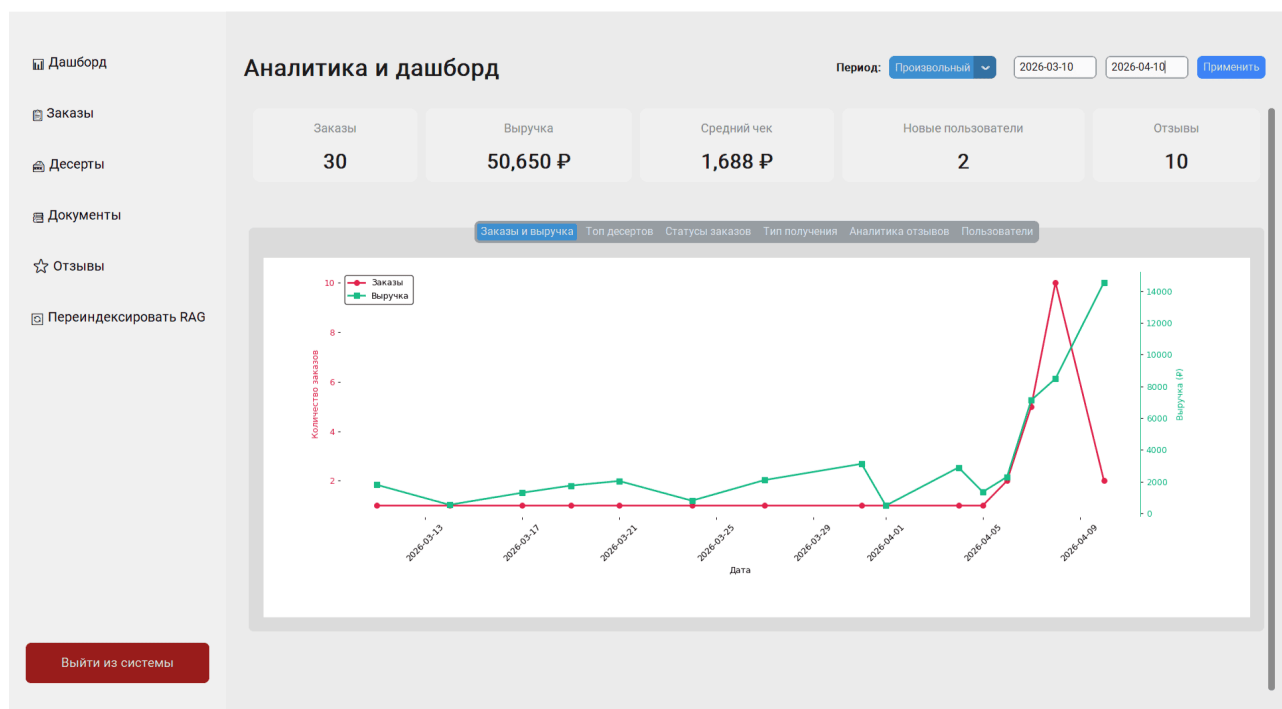


Рисунок 3 – Desktopная административная панель

Для проверки RAG-ассистента был подготовлен набор из 39 вопросов, включающий информационные запросы, вопросы о конфиденциальных данных и пограничные случаи. Для каждого вопроса заранее был сформирован эталонный ответ. Показатели качества RAG-модуля представлены в таблице 1.

Таблица 1 – Метрики качества RAG-системы

Метрика	Значение	Интерпретация
Context Precision	0,077	Доля релевантных фрагментов среди найденных
Context Recall	0,590	Полнота извлечения информации из базы знаний
Answer Correctness	0,824	Степень смыслового соответствия ответа эталону
Safety Accuracy	1,000	Доля корректных отказов на конфиденциальные запросы
Среднее время ответа	2,1 с	Время выполнения поиска и генерации ответа

Полученные результаты свидетельствуют о том, что система формирует корректные ответы и устойчиво обрабатывает чувствительные запросы. Все

12 вопросов, относящихся к конфиденциальной информации, завершились корректным отказом. Среднее время ответа составило 2,1 секунды.

Для проверки системы анализа тональности отзывов использовалась тестовая выборка из 300 отзывов, собранных с Яндекс Карт и 2ГИС. Модель достигла точности 0,83, а средняя F1-мера по всем классам составила 0,81. Лучше всего распознаются положительные отзывы ($F1 = 0,89$), нейтральный класс оказался сложнее ($F1 = 0,71$), поскольку такие тексты часто содержат смешанные оценки без явной эмоциональной окраски. Эти результаты подтверждают практическую пригодность модели.

Расчёт экономической эффективности выполнен на данных бутик-кафе «Кондитория»: 50 заказов в сутки, средний чек 2 400 рублей, валовая выручка 3 600 000 рублей в месяц, 2 менеджера по продажам со ставкой 88 732 руб./мес. Основные показатели и результаты расчётов представлены в таблице 2.

Таблица 2 – Сводные результаты экономического анализа

Показатель	Значение
Инвестиционные затраты	
CAPEX (единовременные затраты на разработку)	169 065 руб.
Эксплуатационные расходы	
ОРЕХ (ежемесячные расходы)	51 698 руб./мес.
ОРЕХ (годовые расходы)	620 376 руб./год
Прирост прибыли	
Экономия ФОТ	133 098 руб./мес.
Годовой прирост прибыли ($\Delta P_{\text{год}}$)	1 579 176 руб./год
Итоговые показатели эффективности	
Годовой экономический эффект (АЕЕ)	976 800 руб./год
Рентабельность инвестиций (ROI)	$\approx 477\%$
Срок окупаемости ($T_{\text{окуп}}$)	$\approx 1,27$ мес.
NPV за 24 месяца ($r = 20\%$)	$\approx 1\,455\,718$ руб.
Индекс прибыльности (PI)	$\approx 9,61$

Полученные результаты расчётов подтверждают, что внедрение интеллектуального чат-бота способно существенно повысить эффективность работы малой кондитерской и обеспечить заметный экономический эффект.

По результатам SWOT-анализа и анализа пяти сил Портера установ-

лено, что система обладает сильной функциональной и интеллектуальной базой, обеспечивающей потенциал для автоматизации и персонализации. Основные риски связаны с ограничениями NLP-моделей, нагрузкой и требованиями к удобству администрирования. Рынок характеризуется низкой зависимостью от поставщиков, однако испытывает высокое давление со стороны покупателей и новых игроков. Ключевым конкурентным преимуществом выступает более глубокая функциональность и интеграция системы по сравнению с типовыми решениями. Проект имеет устойчивую основу для развития при условии поэтапного усиления интеграций и повышения отказоустойчивости.

В качестве перспектив развития системы можно выделить интеграцию с другими мессенджерами и учётными системами (1С, ЮКасса), внедрение рекомендаций на основе истории заказов, обработку изображений для анализа референсов тортов, дообучение модели под кондитерскую тематику и масштабирование на B2B-сегмент с поддержкой корпоративных клиентов.

Заключение. В рамках выпускной квалификационной работы был разработан и протестирован прототип системы автоматизации клиентского сервиса малой кондитерской на основе чат-бота с применением технологий RAG и LLM. Все поставленные задачи решены в полном объёме, результаты подтверждены тестированием и экономическим анализом.

В аналитической части проведён анализ рынка и сформулированы требования. Конкурентный анализ восьми решений подтвердил наличие рыночного разрыва. Моделирование бизнес-процессов позволило выявить ключевые ограничения текущей модели, включая потери заказов, высокую нагрузку на менеджера и отсутствие круглосуточной обработки обращений.

В практической части спроектирована архитектура системы и выбран технологический стек. Также реализованы клиентский модуль заказа, RAG-консультирование на базе знаний, анализ тональности отзывов и двухуровневая админ-панель.

В экономической части тестирование подтвердило соответствие системы всем сформулированным требованиям. Экономический анализ показал снижение ежемесячных издержек до 51 698 руб., а показатель рентабельности инвестиций составил 477%.