

**Федоров Владимир Александрович**

*студент, Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Восточно-Сибирский государственный университет технологий и  
управления»,  
Россия, г. Улан-Удэ*

*Научный руководитель:*

**Евдокимова Инга Сергеевна**

*кандидат технических наук, доцент,  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Восточно-Сибирский государственный университет технологий и  
управления»,  
Россия, г. Улан-Удэ*

## **РАЗРАБОТКА TELEGRAM-БОТА КАК КАНАЛА ВЗАИМОДЕЙСТВИЯ ИНТЕЛЛЕКТУАЛЬНОГО АССИСТЕНТА С КЛИЕНТАМИ КОМПАНИИ**

**Аннотация:** В статье рассматривается разработка Telegram-бота как интерфейсного канала интеллектуальной системы поддержки клиентов. Описывается архитектура бота, реализованного на основе библиотеки `python-telegram-bot`, включая организацию диалоговых сценариев, управление историей переписки и логику автоматического перенаправления запросов. Особое внимание уделяется механизму выявления намерения клиента совершить покупку на основе анализа свободного текста, а также решениям инженерных проблем, связанных со спецификой платформы Telegram. Показано, что использование популярного мессенджера в качестве канала взаимодействия

позволяет значительно повысить доступность системы поддержки для конечных пользователей.

**Ключевые слова:** Telegram Bot API, python-telegram-bot, чат-бот, ConversationHandler, обработка естественного языка, распознавание намерений, автоматизация поддержки клиентов, мессенджер-маркетинг, перенаправление лидов.

## **Введение**

Мессенджер Telegram является одной из наиболее популярных платформ обмена сообщениями в России и странах СНГ: по данным открытых источников, в 2025 году ежемесячная аудитория приложения превысила 950 миллионов пользователей по всему миру [1]. Широкое распространение мессенджера, развитый Bot API и возможность создания интерактивных интерфейсов с кнопками и меню делают Telegram привлекательной платформой для развёртывания корпоративных чат-ботов.

В контексте автоматизации технической поддержки Telegram-бот выполняет функцию универсального клиентского интерфейса: пользователи могут задавать вопросы в привычной среде мессенджера, не устанавливая дополнительных приложений и не открывая браузер. При этом бот способен не только отвечать на вопросы по документации, но и выявлять коммерческие намерения клиентов и своевременно передавать их в отдел продаж.

Цель настоящей работы — описать архитектуру и ключевые технические решения, применённые при разработке Telegram-бота в составе интеллектуальной системы поддержки клиентов программной компании.

### **1. Архитектура Telegram-бота и выбор инструментов разработки**

Разработка бота велась на языке Python 3.11 с использованием библиотеки python-telegram-bot версии 20.x, поддерживающей асинхронную работу на основе asyncio. Выбор данной библиотеки обусловлен её зрелостью, подробной документацией и поддержкой всех актуальных возможностей Telegram Bot API, включая ConversationHandler для управления многошаговыми диалогами.

Взаимодействие бота с серверной частью системы организовано через REST API, реализованный на FastAPI. Telegram-бот выступает клиентом по отношению к бэкенду: при получении вопроса от пользователя бот формирует HTTP-запрос к эндпоинту /ask, передаёт текст сообщения и идентификатор пользователя, а затем возвращает полученный ответ обратно в чат. Такое разделение позволяет независимо масштабировать компоненты системы.

Бот работает в режиме polling (опрос сервера Telegram с заданным интервалом) и запускается как самостоятельный сервис в контейнере Docker, что обеспечивает его стабильную работу и удобное управление жизненным циклом.

## **2. Интерфейс и диалоговые сценарии**

Пользовательский интерфейс бота организован через систему inline-кнопок, формирующих постоянное меню в нижней части экрана чата. Меню содержит четыре пункта: «Документация» (переход в режим ответов на технические вопросы), «Купить / Лицензия» (прямое перенаправление в отдел продаж), «Позвать специалиста» (эскалация к живому сотруднику) и «О боте» (информация о системе). Наличие структурированного меню снижает порог входа для пользователей и ускоряет навигацию по функциям.

Помимо меню, бот обрабатывает свободный текст сообщений пользователя в режиме реального времени. При получении каждого сообщения система последовательно проверяет: является ли сообщение командой меню;

содержит ли оно признаки намерения купить продукт; либо является техническим вопросом, требующим поиска по документации.

История диалога сохраняется для каждого пользователя в памяти приложения в виде словаря, ключом которого служит идентификатор пользователя Telegram. При передаче диалога специалисту или в отдел продаж полная история переписки передаётся вместе с заявкой, что позволяет сотруднику понять контекст обращения без необходимости повторного расспроса клиента.

### **3. Механизм выявления намерения совершить покупку**

Одной из ключевых функций бота является автоматическое выявление коммерческого намерения клиента — ситуации, когда пользователь заинтересован в приобретении продукта или лицензии. Своевременная передача такого клиента в отдел продаж имеет непосредственное влияние на конверсию и выручку компании.

Для решения данной задачи была реализована функция `is_sales_intent()`, выполняющая лексический анализ входящего сообщения. Функция проверяет наличие в тексте более 25 триггерных слов и словосочетаний, характерных для намерения купить: «купить», «приобрести», «стоимость», «цена», «прайс», «лицензия», «лицензирование», «коммерческое», «заказать», «оформить» и ряд других, включая составные фразы типа «сколько стоит» и «хочу купить».

Анализ проводится после приведения текста к нижнему регистру и удаления лишних пробелов, что обеспечивает корректную работу вне зависимости от регистра написания и опечаток в конце слова. При обнаружении намерения бот незамедлительно сообщает пользователю о передаче его запроса в отдел продаж, отображает контактную информацию менеджера и

прикладывает полную историю диалога. Параллельно система может формировать уведомление для CRM-системы.

Следует отметить, что выбор в пользу лексического анализа, а не классификатора на основе машинного обучения, был обусловлен требованием полного локального развёртывания системы и необходимостью обеспечить прозрачность и предсказуемость поведения. Лексический подход легко поддерживается: для добавления нового триггерного выражения достаточно дополнить соответствующий список.

#### **4. Инженерные проблемы и пути их решения**

В процессе разработки бота был выявлен и решён ряд специфических технических проблем, связанных с особенностями платформы Telegram.

Первая проблема — несовместимость разметки Markdown с форматом MarkdownV2, требуемым Telegram Bot API. Языковая модель генерирует ответы с использованием стандартного синтаксиса Markdown (звёздочки для жирного шрифта, знаки подчёркивания для курсива), тогда как Telegram требует экранирования этих и ряда других спецсимволов с помощью обратного слеша. При несоблюдении данного требования API возвращает ошибку 400 (Bad Request) и сообщение не отправляется.

Решением стала функция `escape_markdown()`, которая выполняет экранирование всех специальных символов MarkdownV2 перед отправкой. Дополнительно был реализован механизм `fallback`: если даже после экранирования отправка не удаётся (например, из-за некорректной вложенности тегов), бот повторно отправляет то же сообщение в режиме `plain text` без какого-либо форматирования. Такой двухуровневый подход полностью исключает потерю ответа.

Вторая проблема — отображение ссылок на источники в ситуациях, когда система не нашла релевантного ответа в базе знаний и инициировала эскалацию. В первоначальной реализации бот всегда прикладывал ссылки на документацию, даже если ответ был сформирован как сообщение об эскалации. Решением стала двойная проверка: анализ флага `escalate`, возвращаемого бэкендом, а также разбор текста ответа на наличие фраз-маркеров отказа. Ссылки на источники отображаются только при одновременном отсутствии обоих признаков эскалации.

## **5. Анализ результатов эксплуатации**

По данным встроенного дашборда статистики, в ходе тестовой эксплуатации Telegram-бот обработал 28 диалогов. Из них 4 обращения были эскалированы специалисту ввиду отсутствия ответа в базе знаний. Среди обработанных запросов преобладали вопросы об установке и обновлении программного обеспечения, составившие около 60% от общего числа обращений.

Отдельного внимания заслуживает пользовательский опыт: в отличие от веб-чата, для работы с Telegram-ботом пользователю не нужно открывать браузер — достаточно перейти по ссылке или QR-коду прямо в мобильном приложении мессенджера. Это существенно снижает порог входа для мобильной аудитории и делает поддержку более доступной.

Механизм выявления намерений продемонстрировал корректную работу: ни один тестовый сценарий с запросом цены или лицензии не привёл к попытке поиска по документации — все подобные запросы были корректно перенаправлены в отдел продаж.

## **Заключение**

В ходе работы был разработан и протестирован Telegram-бот в составе интеллектуальной системы поддержки клиентов. Ключевыми результатами

являются: реализация полноценного диалогового интерфейса с постоянным меню и историей переписки; создание надёжного механизма выявления коммерческих намерений клиентов; решение платформенно-специфических инженерных проблем, связанных с форматированием сообщений Telegram.

Практика показала, что Telegram является эффективным каналом для развёртывания корпоративных ИИ-ассистентов: платформа обеспечивает широкий охват аудитории, предоставляет богатые возможности для построения интерактивных интерфейсов и не требует от пользователей установки дополнительного программного обеспечения. Перспективными направлениями развития являются поддержка голосовых сообщений, интеграция с CRM-системами в реальном времени и персонализация ответов на основе истории взаимодействия с конкретным пользователем.

### **Список литературы**

1. Telegram: Статистика мессенджера в 2025 году [Электронный ресурс]. — Режим доступа: <https://telegram.org/blog> (дата обращения: 15.04.2026).
2. python-telegram-bot Documentation [Электронный ресурс]. — Режим доступа: <https://docs.python-telegram-bot.org/> (дата обращения: 10.04.2026).
3. Telegram Bot API Reference [Электронный ресурс]. — Режим доступа: <https://core.telegram.org/bots/api> (дата обращения: 10.04.2026).
4. Shawar B. A., Atwell E. Chatbots: Are they Really Useful? // LDV Forum. — 2007. — Vol. 22, № 1. — P. 29–49.
5. Adamopoulou E., Moussiades L. An Overview of Chatbot Technology // Artificial Intelligence Applications and Innovations. — Springer, Cham. — 2020. — P. 373–383.
6. Xu A., Liu Z., Guo Y. et al. A New Chatbot for Customer Service on Social Media // Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. — 2017. — P. 3506–3510.

7. FastAPI Documentation [Электронный ресурс]. — Режим доступа: <https://fastapi.tiangolo.com/> (дата обращения: 10.04.2026).