УДК 004.42

TELEGRAM-БОТ КАК ИНСТРУМЕНТ АВТОМАТИЗАЦИИ ДОСТУПА СТУДЕНТОВ К БАЗЕ ДАННЫХ МЕСТ ПРАКТИК В КУЗГТУ

Салмин А.А. – студент ИИТМА,
Мельников Н.А. – студент ИИТМА,
Пимонов А.Г. – научный руководитель, д.т.н., профессор
Кузбасский государственный технический университет имени Т.Ф. Горбачева
Россия, г. Кемерово

Проблема эффективного информационного взаимодействия между университетом и студентами во многом определяет качество образовательной среды. На протяжении долгого времени основным инструментом распространения сведений о местах практики выступали статические веб-страницы и бумажные информационные стенды. Однако современные обучающиеся активно используют мобильные устройства, предпочитая инструменты, доступные в простой, привычной и быстрой форме. Это приводит к росту интереса к интеграции сервисов [1, 2] на базе мессенджеров.

Telegram является одной из наиболее удобных и технологически развитых платформ для создания ботов, сочетая скорость, доступность, широкую аудиторию и открытый API. Пользовательский интерфейс Telegram стал стандартом коммуникации для молодёжи: простота меню, мгновенные уведомления, отсутствие необходимости устанавливать отдельные приложения – всё это делает платформу подходящим инструментом для работы с большим количеством пользователей [2].

Современные технологии разработки позволяют создавать программные решения, интегрирующиеся в повседневное пространство студента. Чат-боты способны выполнять функции информационных справочников, навигаторов и помощников, обеспечивая моментальный доступ к актуальной информации о практике, мероприятиях, образовательных программах. Такие системы играют всё более важную роль в цифровой трансформации университетов [4].

Telegram-бот представляет собой программный комплекс, функционирующий в рамках мессенджер-платформы и реализующий взаимодействие через естественный для пользователя интерфейс чата [1]. Его архитектурные особенности позволяют интегрировать различные источники данных, обеспечивая при этом высокую степень автономности и масштабируемости решения. Такие системы демонстрируют принципиально новые характеристики доступности и удобства использования по сравнению с традиционными веб-приложениями, особенно в контексте мобильного доступа к информации [2].

Telegram-бот для КузГТУ (рис. 1) был создан как инструмент автоматизации доступа студентов к списку мест прохождения практики. Основная цель заключалась в том, чтобы устранить разрозненность информации,

минимизировать ручные операции сотрудников и предоставить студентам возможность быстрого поиска подходящих вариантов [3].

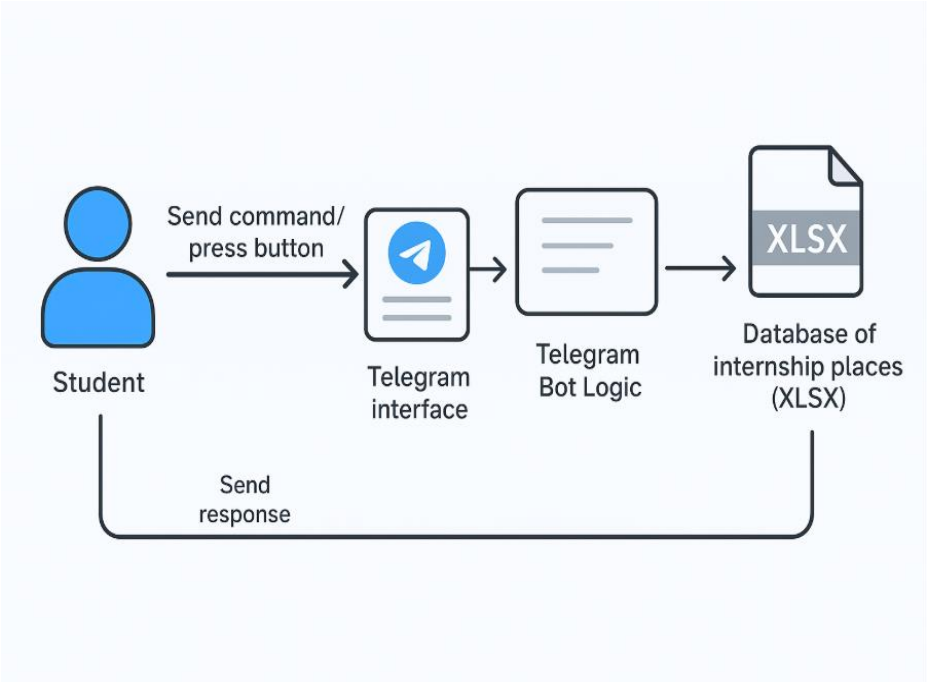

Рисунок 1 – Архитектура работы Telegram-бота для доступа к базе данных мест практик

Работа над проектом началась с определения ключевых требований. Были изучены запросы студентов, особенности взаимодействия с отделами практик, форматы представления данных. Анализ показал, что студентам важна не только информация о компании, но и структурированность: направление подготовки, адрес, контакты, перечень задач на практике. Особое внимание уделялось тому, чтобы взаимодействие занимало не более 15-20 секунд. Следующим этапом стало проектирование структуры данных. На основе анализа вакансий была подготовлена таблица формата XLSX, в которую были включены сведения о направлениях подготовки, предприятиях, их адресах, описания практик и контактных лицах. Данный формат удобен тем, что любые изменения могут быть выполнены без программных знаний, что снижает требования к администрированию. Интерактивная логика бота, часть кода которого приведена ниже, моделировалась с учётом необходимости устойчивой работы при высоких нагрузках. На этапе разработки создавались прототипы меню, варианты ответов, тестовые сценарии. При проектировании интерфейса большое внимание уделялось тому, чтобы студент мог выполнить целевое действие минимальным количеством шагов.

```
import requests
import time
import json
import pandas as pd
```

```
TOKEN = "7261176930:AAFvqzNRFKeeRRqVWjhU5VA9-ObyqcvOcbk"
EXCEL_FILE = "kuzstubasa.xlsx"
def load_data():
    df = pd.read_excel(EXCEL_FILE)
    data = {}
    for direction in df["Направление"].unique():
        orgs = df[df["Направление"] ==
                          direction]["Организация"].tolist()
        data[direction] = orgs
    return data
DATA = load_data()
def get_updates(offset=None):
    url = f"https://api.telegram.org/bot{TOKEN}/getUpdates"
    params = {"timeout": 100, "offset": offset}
    return requests.get(url, params=params).json()
def send_message(chat_id, text, reply_markup=None):
    url = f"https://api.telegram.org/bot{TOKEN}/sendMessage"
    params = {"chat_id": chat_id, "text": text}
    if reply_markup:
        params["reply_markup"] = json.dumps(reply_markup)
    requests.post(url, params=params)
def get_main_keyboard():
    keyboard = {
        "keyboard": [[{"text": direction}] for direction in DATA.keys()],
        "resize_keyboard": True,
        "one_time_keyboard": False
    }
    return keyboard
def main():
    last_update_id = None
    while True:
        updates = get_updates(last_update_id)
        if "result" in updates:
            for update in updates["result"]:
                last_update_id = update["update_id"] + 1
                if "message" in update and "text" in update["message"]:
                    chat_id = update["message"]["chat"]["id"]
                    text = update["message"]["text"]
                    if text == "/start":
                        send_message(chat_id, "👋 Добро пожаловать!
                        Выберите направление:", get_main_keyboard())
                        continue
                    if text in DATA:
                        for org in DATA[text]:
```

```
                send_message(chat_id, f"🏢 {org}")
            continue
    time.sleep(1)
```

После завершения проектирования началась стадия реализации. Логика диалога, структуры меню и фильтрации данных была построена с использованием модульного принципа. Это позволило при необходимости добавлять новые направления подготовки или функциональные блоки без переписывания основной части кода. Особой задачей являлось обеспечение корректной работы с объектами Excel. Были разработаны механизмы проверки корректности данных, автоматического обновления и защиты от ошибочного заполнения. В рамках предварительной проверки работоспособности бот был протестирован группой из более чем 100 студентов разных направлений. Это позволило выявить ошибки в структуре меню, оптимизировать формулировки, улучшить читаемость данных. Тестирование подтвердило возможность бота обрабатывать высокий поток запросов в течение короткого времени и корректно работать с различными сценариями поведения пользователей.

Отдельное внимание при разработке уделялось вопросам защиты данных. Поскольку бот работает с информацией для студентов, но не обрабатывает персональные данные, он не требует хранения закрытых сведений. Однако перспективным направлением была признана идея внедрения системы авторизации по студенческому билету. Данное внедрение авторизации предполагается в будущей версии проекта.

Telegram был выбран не случайно. Хотя существуют и другие платформы, такие как ВКонтакте, WhatsApp или веб-порталы, каждое из решений имеет ограничения. ВКонтакте ограничивает использование ряда функций бота без официального приложения; WhatsApp фактически закрыт для полноценных ботов; веб-порталы требуют отдельного интерфейса и не обеспечивают быстроты. Telegram сочетает все преимущества: простоту доступа, стабильный API, мобильность и возможность расширения функций [1]. Архитектура бота реализована по трехзвенной модели:

1) *уровень представления* отвечает за взаимодействие с пользователем через интерфейс чата, включая кнопки и меню;

2) *уровень бизнес-логики* обрабатывает команды пользователя, управляет сценариями диалога (Finite-State Machine) и формирует запросы к уровню данных;

3) *уровень доступа к данным* выполняет операции чтения и фильтрации информации. В качестве основного источника данных использовался структурированный файл в формате XLSX, что позволило быстро интегрировать решение без необходимости развертывания полноценной системы управления базами данных. Для работы с файлом применялись библиотеки Pandas и Openpyxl.

В результате был создан и запущен в эксплуатацию Telegram-бот (рис. 2), предоставляющий следующий функционал:

1. просмотр списка направлений практик;

2. получение детальной информации по выбранной вакансии, включая контакты работодателя и ссылки, если таковые имеются.

Анализ работы системы показал, что среднее время отклика на запрос пользователя не превышает 1-2 секунд, а процесс поиска и фильтрации вакансий занимает у студента в среднем 1-2 минуты, что существенно быстрее традиционных методов [4]. Использование файла XLSX в качестве бэкенда доказало свою эффективность для прототипирования и внедрения, обеспечивая простоту администрирования и обновления базы данных ответственным сотрудником вуза.

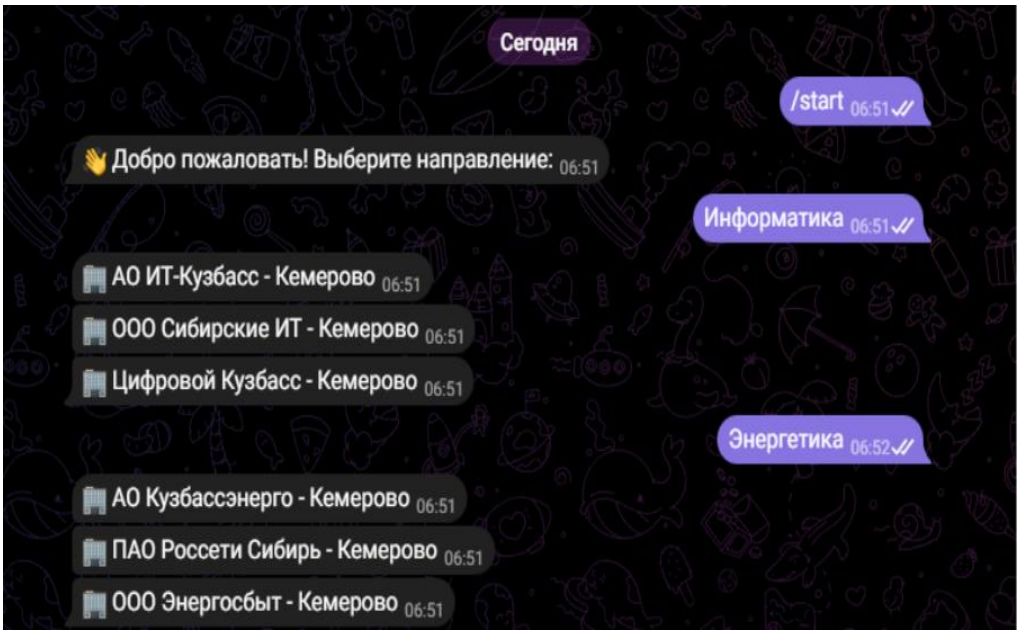

Рисунок 2 – Пример работы Telegram-бота для поиска мест практик

В заключении хотелось бы добавить, что реализация проекта по разработке системы автоматизации доступа к базе мест практик для студентов КузГТУ на основе Telegram-бота представляет собой значимый вклад в цифровизацию образовательной среды университета. Проект направлен на решение актуальной задачи оптимизации процессов взаимодействия между студентами и рынком труда, обеспечивая централизованный доступ к вакансиям практик и стажировок.

Ключевые преимущества проекта:

1) создание интуитивно понятного сервиса для студентов всех курсов и направлений подготовки;

2) снижение административной нагрузки на отделы практик и карьерные центры;

3) формирование единого информационного пространства для трудоустройства студентов;

4) развитие профессиональных компетенций участников проекта в области full-stack разработки.

Перспективы развития проекта предусматривают переход от прототипа в формате Telegram-бота к созданию полнофункциональной информационной системы, включающей:

1) разработку кроссплатформенного мобильного приложения с расширенным функционалом;
2) создание веб-платформы для администраторов и работодателей;
3) интеграцию с внешними HR-системами и порталами вакансий.

Реализация этих направлений будет осуществляться при поддержке кафедры прикладных информационных технологий КузГТУ и в партнерстве с IT-компаниями региона. Дальнейшее развитие проекта способно трансформировать его в комплексное решение для образовательно-карьерного взаимодействия, что соответствует современным трендам цифровизации высшего образования и подготовки конкурентоспособных специалистов. Проект обладает значительным потенциалом масштабирования и может стать основой для создания университетской экосистемы трудоустройства, укрепляя связи между образовательным учреждением, студентами и работодателями.

Список литературы:

1. Шумилина, М. А. Разработка чат-бота на языке программирования Python в мессенджере «Telegram» [Электронный ресурс] / М. А. Шумилина // Научные известия. – 2022. – № 28. – URL: https://cyberleninka.ru/article/n/razrabotka-chat-bota-na-yazyke-programmirovaniya-python-v-messendzhere-telegram (дата обращения: 17.11.2025).

2. Тищенко, А. А. Разработка чат-бота в качестве виртуального помощника для мотивации студентов к получению профессии [Электронный ресурс] / А. А. Тищенко // Эргодизайн. – 2021. – № 2 (12). – URL: https://cyberleninka.ru/article/n/razrabotka-chat-bota-v-kachestve-virtualnogo-pomoschnika-dlya-motivatsii-studentov-k-polucheniyu-professii (дата обращения: 17.11.2025).

3. Токарев, Э. А. Структурирование обработчиков в Telegram-ботах для оптимизации взаимодействия [Электронный ресурс] / Э. А. Токарев // Вестник науки. – 2024. – № 5 (74). – URL: https://cyberleninka.ru/article/n/strukturirovanie-obrabotchikov-v-telegram-botah-dlya-optimizatsii-vzaimodeystviya (дата обращения: 17.11.2025).

Валинурова, А. А. Алгоритм разработки Telegram-бота – продуктивного помощника современного бизнеса [Электронный ресурс] / А. А. Валинурова // Современные наукоемкие технологии. Региональное приложение. – 2023. – № 2 (74). – URL: https://cyberleninka.ru/article/n/algoritm-razrabotki-telegram-bota-produktivnogo-pomoschnika-sovremennogo-biznesa (дата обращения: 17.11.2025).