

УДК 004.04

## **ПРИМЕНЕНИЕ ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦИИ ЗАПОЛНЕНИЯ ПЕРСОНАЛЬНЫХ ДАННЫХ СТУДЕНТА ДЛЯ УЧЕБНЫХ ЗАВЕДЕНИЙ**

Кабанов И.А., студент гр. ИТб-202, IV курс  
Научный руководитель Николаев П.И., к.н., доцент  
Кузбасский государственный технический  
университет имени Т. Ф. Горбачева г. Кемерово

В нашем веке технологии играют важную роль в различных сферах жизни, включая образование. Учебным заведениям необходимо эффективно проводить сбор и хранение информации о своих студентах. Одним из важных моментов будет сбор персональных данных, таких как контактная информация и другие данные.

Обычные методы сбора данных, такие как заполнение бумажных анкет или собственноручный ввод информации в электронные формы на веб-сайтах учебных заведений, оказываются неудобными и сложным для студентов. Такие подходы затрачивают время и ресурсы для обработки и хранения данных со стороны администрации учебного заведения.

Поэтому возникает необходимость в поиске новых решений для оптимизации этого процесса. Одно из решений будет использование чат-бота - программных агентов, взаимодействующих с пользователями через текстовые сообщения.

В этой статье показана реализация и применение чат-бота для автоматизации процесса заполнения персональных данных студентов в учебных заведениях. Рассмотрим плюсы такого решения, процесс разработки и внедрения чат-бота. Этот чат-бот взаимодействует с пользователями в естественной форме через текстовые сообщения, собирать необходимые данные и автоматически сохранять их в базу данных учебного заведения. Этот подход позволяет упростить и ускорить процесс заполнения всех данных, а также уменьшается вероятность ошибок.

Чат-бот будет автоматически работать в мессенджере Telegram, он написан на языке Python при помощи интегрированной среды разработки Visual Studio, название нашего чат-бота "StudentAdminBot". Также в систему будет входить база данных (БД), к которой будет обращаться бот, созданная в СУБД Microsoft SQL Server, и хранящая данные. Программа, отвечающая за автоматическую работу чат-бота, будет работать на сервере постоянно, чтобы взаимодействовать с пользователем. Чат-бот будет доступен для пользователей в Telegram. Системе "StudentAdminBot" доступна возможность отображения всех персональных данных студентов

на сайтах учебного заведения. На сайте будут доступны функции просмотра данных студента, можно просматривать персональные данные студентов, такие как ФИО, контактная информация и другие, так же может осуществляться поиск студентов, что упрощает работу с базой данных, обеспечивает быстрый доступ к нужной информации, добавленную информацию можно будет изменять и вносить новые данные, если это необходимо. Эти функции позволяют поддерживать актуальность информации и корректировать ошибки, как и чат-бот избавляют сотрудников от лишней работы.

Диаграмма деятельности процесса заполнения данных в телеграмм боте для отображения на сайте изображена на рисунке 1.

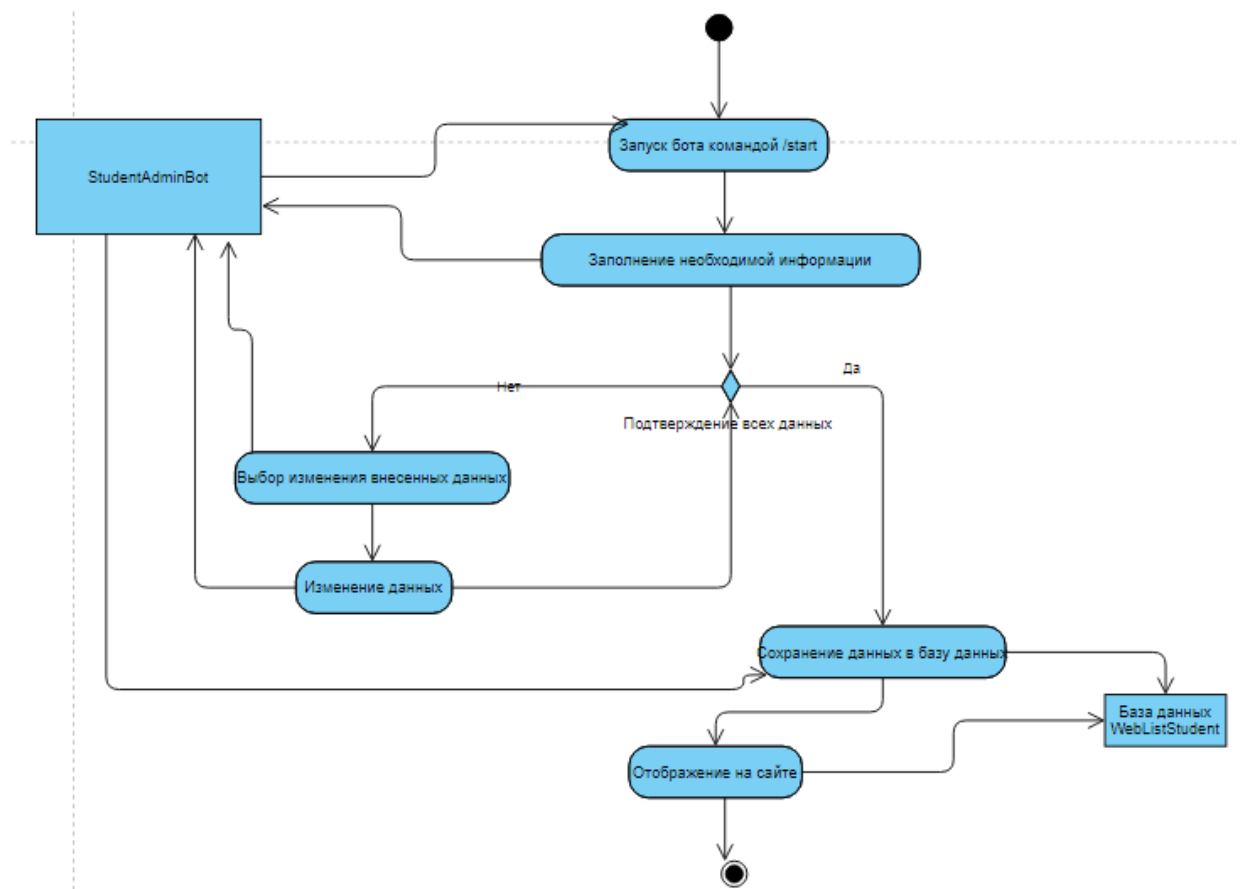


Рисунок 1 – Диаграмма деятельности процесса заполнения данных в телеграмм боте для отображения на сайте.

Чат-бот взаимодействует со студентом, запрашивая у него все необходимые персональные данные, такие как ФИО, контактная информация и другие. По окончании заполнения всех данных студенту будут предоставлены все эти данные в чате, для последующего подтверждения и сохранения в базе данных. Студенты смогут в любое время изменить свою информацию и удалить прошлую свою запись, обратившись к чат-боту и указав команду `/delete_data`, с последующим новым заполнением своих данных и их сохранением. Для того чтобы удались данные именно этого пользователя, при заполнении данных и

сохранении, ему присваивается идентификатор, чтобы при удалении обращаться именно к этому значению и связанными с ним данными. Диаграмма деятельности для идентификации пользователя представлена на рисунке 2.

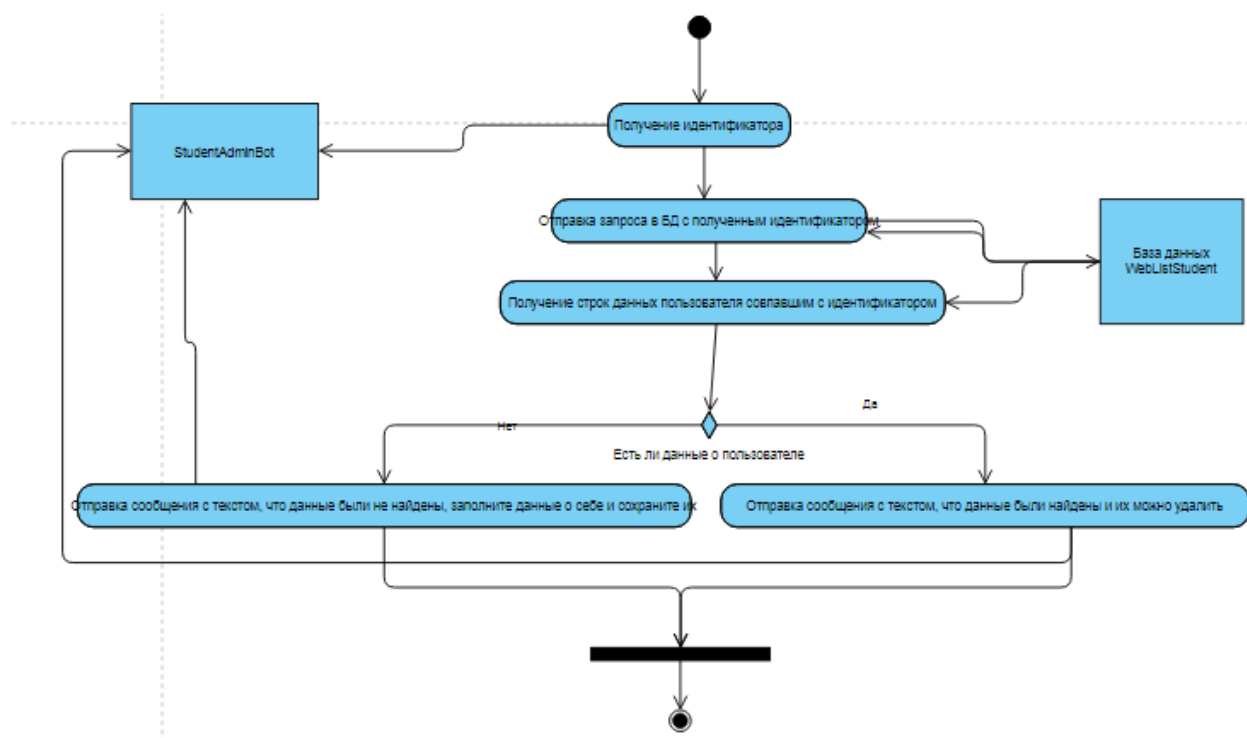


Рисунок 2 – Диаграмма деятельности процесса идентификации пользователя.

Для создания чат-бота, сначала необходимо зарегистрировать его в Telegram. Вся процедура регистрации проходит через официального бота – BotFather. С помощью BotFather задается название бота, которое будет отображаться пользователям, уникальное имя для поиска бота, заканчивающееся на "bot", наш @StudentAdminBot\_bot. Взаимодействие с BotFather, как и с любым другим чат-ботом, осуществляется при помощи команд. Некоторые из доступных команд в BotFather включают:

- /newbot – создание нового чат-бота;
- /setname – изменение названия чат-бота;
- /setcommands – указание списка доступных команд для бота;
- /setuserpic – изменение изображения профиля чат-бота;
- /setabouttext – добавление информации о боте;
- /token – получение токена для авторизации бота.

После всех процедур, а именно назначения названия и уникального имени бота, мы получаем токен – уникальную последовательность символов, которая позволит управлять ботом через нашу программу. Важно сохранять токен в полной конфиденциальности. Созданный бот

представлен на рисунке 3.

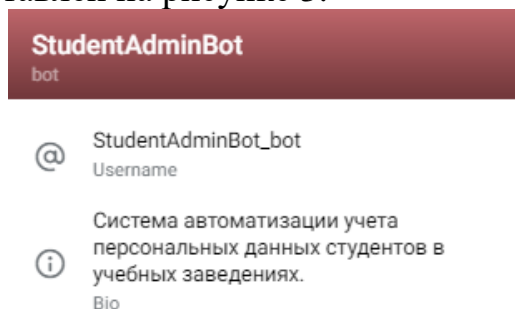


Рисунок 3 – Созданный чат-бот в telegram.

После получения токена переходим к разработке программы, которая будет взаимодействовать с чат-ботом. Для этого используем среду разработки Visual Studio и устанавливаем необходимый пакет Telegram.Bot через систему управления пакетами NuGet. Этот пакет предоставляет удобные инструменты для работы с API Telegram.

Приложение должно быть размещено на сервере, где будет работать непрерывно. Это приложение будет получать обновления от чат-бота через метод bot.polling. Обновления могут содержать информацию о входящих сообщениях от пользователей, командах и других событиях, происходящих в чате. После получения обновлений от чат-бота, приложение должно обработать их соответствующим образом. Например, если пользователь отправил сообщение, приложение должно обработать это сообщение и выполнить соответствующие действия, например, отправить ответное сообщение или выполнить определенную команду. Подключение бота к базе данных и подключение токена на рисунке 4

```
• Строка подключения
conn_str = f'DRIVER={{SQL Server}};SERVER={server};DATABASE={database};Trusted_Connection=yes'

• Создание экземпляра бота
bot = telebot.TeleBot('7029605975:AAFwC2aLa3xcqdPAN')
```

Рисунок 4 – Подключение к чат-боту.

Определим, на какие команды будет отвечать чат-бот. Будет всего 2 команды: /start - отправляется боту при для начала диалога. В ответ бот отправляет приветствие и запрашивает данные для заполнения. /delete\_data - отправляется пользователем для удаления заявки. В ответ пользователю приходит сообщение, что данные удалены или не найдены. Пример обработки команды /start с последующими обработками для заполнения данных показан на рисунке 5.

```
# Обработчик команды /start
@bot.message_handler(commands=['start'])
def handle_start(message):
    bot.send_message(message.chat.id, "Привет! Давайте начнем.")

# Обработчик ввода названия страны
@bot.message_handler(func=lambda message: True)
def handle_country(message):
    user_data['country'] = message.text
    bot.send_message(message.chat.id, "Пожалуйста, введите название страны:")
    bot.register_next_step_handler(message, handle_specialization)
```

Рисунок 5 – Обработка команды /start с последующими обработками для заполнения данных.

Для системы "StudentAdminBot" важна функция для подтверждения введенных данных, для обеспечения корректности данных, введенных пользователем в чат-боте. Она предоставляет пользователю возможность просмотра введенной информации перед ее сохранением в базу данных. Это позволяет избежать ошибок в данных и их дальнейшей корректировки. После ввода пользовательской информации, чат-бот отправляет сообщение с подтверждением, содержащим введенные данные. Пользователь имеет возможность проверить эту информацию и подтвердить ее, отправив 'да', либо запросить изменение, отправив 'нет'. В случае отрицательного ответа, бот предоставляет пользователю возможность выбрать, какое поле данных нужно изменить. Метод подтверждения данных и обработчик для ответа на подтверждение показан на рисунке 6.

```
# Функция для подтверждения введенных данных
def confirm_data(message):
    confirmation_text = f"Страна: {user_data['country']}\n" \
        f"Специализация: {user_data['specialization']}\n" \
        "Проверьте все данные!"
    bot.send_message(message.chat.id, confirmation_text)
    bot.send_message(message.chat.id, "Все данные верны? Отправьте 'да' или 'нет'.")
    bot.register_next_step_handler(message, handle_confirmation_response)

# Обработчик для ответа на подтверждение
def handle_confirmation_response(message):
    response = message.text.lower()
    if response == 'да':
        save_data(message)
        bot.send_message(message.chat.id, "данные успешно добавлены.")
        bot.send_message(message.chat.id, "Если заявка была не правильно заполнена, то чтобы её удалить, отправьте или нажмете команду /delete_da")
    elif response == 'нет':
        bot.send_message(message.chat.id, "Что вы хотите изменить:\n1 - страну\n2 - специализацию\nотправьте '1' или '2'?")
        bot.register_next_step_handler(message, handle_change_selection)
    else:
        bot.send_message(message.chat.id, "Пожалуйста, ответьте 'да' или 'нет'.")
```

Рисунок 6 – Функция для подтверждения введенных данных и обработчик для ответа на подтверждение.

Тестирование работы чат-бота «StudentAdminBot» показано на рисунке 7.



Рисунок 7 – Тестирование чат-бота «StudentAdminBot».

Данные загружаются в бд и отображаются на сайте, отображение показано на рисунке 8.

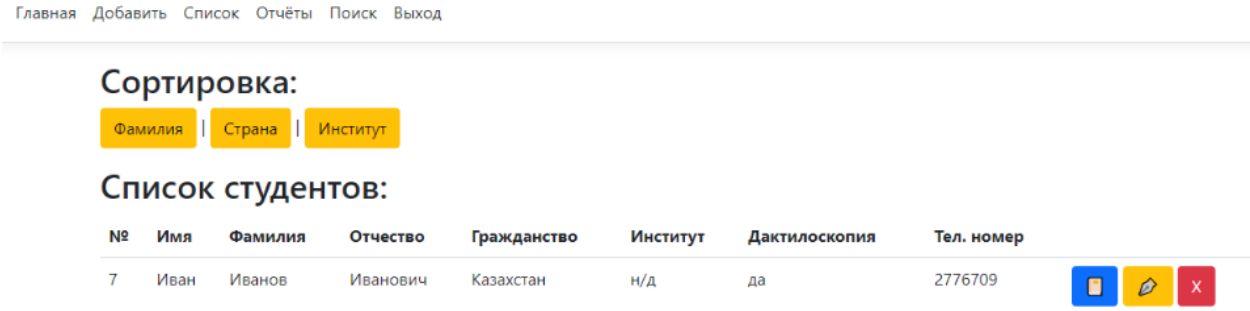


Рисунок 8 – Отображение данных в базе данных.

После внедрения чат-бота в деятельность учебных заведений будет значительное улучшение в эффективности и удобстве обработки персональных данных студентов. Чат-бот "StudentAdminBot" предоставляет простой и быстрый способ заполнения данных студентов, сокращая затраты со временем сотрудников и сокращает возникновение ошибок в данных. Студенты теперь смогут заполнять свои персональные данные непосредственно через мессенджер Telegram. Благодаря этому процессу, сотрудникам учебных заведений не требуется тратить время на ручной ввод и обработку информации, это позволяет сосредоточиться на более важных задачах.

**Список литературы:**

1. Справочник по Telegram Bot API. [Электронный ресурс] URL: <https://tlgrm.ru/docs/bots/api>.
2. Бот telegram на Python. [Электронный ресурс] URL: <https://habr.com/ru/articles/732136>.
3. Моделирование на языке UML в среде Visual Paradigm 14. [Электронный ресурс] – URL: <http://sp.cs.msu.ru/ooap/exer2017.html#exer34>.
4. Работа баз данных с телеграмм ботами. [Электронный ресурс] URL: <https://habr.com/ru/articles/552788>.