

РАЗРАБОТКА CRM-СИСТЕМЫ В TELEGRAM

С.О. Бучнев¹, Я.А. Афанасова¹, Н.Н. Литвинов¹

¹ФГБОУ ВО «Воронежский государственный лесотехнический университет имени Г. Ф. Морозова»

Аннотация. В данной работе при разработке CRM-системы в Telegram был проведен анализ проблемы организации и определен функционал. Боту необходимо: делить на роли, организовывать очередь и распределение вопросов и собирать статистику. База данных спроектирована и разработана в PostgreSQL. Программная составляющая системы рассмотрена на языке Python относительно основного функционала.

Ключевые слова: Telegram, CRM-система, Бот, Программирование, Python.

CRM-SYSTEM DEVELOPING IN TELEGRAM

S.O. Buchnev¹, Y.A. Afanasova¹, N.N. Litvinov¹

¹Voronezh State University of Forestry and Technologies named after G.F. Morozov

Abstract. In this work, when developing a CRM system in Telegram, an analysis of the organization's problem was carried out and the functionality was determined. The bot needs to: divide into roles, organize the queue and distribution of questions, and collect statistics. The database is designed and developed in PostgreSQL. The software component of the system is considered in Python with respect to the basic functionality.

Keywords: Telegram, CRM-system, Bot, Programming, Python.

Telegram – самый популярный мессенджер, благодаря которому возможно общаться с пользователем на другом конце страны. Telegram позволяет отправлять многие типы данных: изображения, видеофайлы, файлы, местоположение, и т.д. Помимо обычных пользователей в Telegram существуют боты.

Боты позволяют расширить возможности мессенджера, так как выполняют заранее прописанные действия по сценарию. Например, можно организовать магазин в Telegram с выводом полноценного ассортимента товаров и проведением

оплаты. Для управления ботом используют API, для которого создается токен с уникальным набором символов. Все API работают по одному принципу: обработка обновлений (отправка сообщений, редактирование, удаление и т.д.) и реагирование на них. Бот способен обрабатывать несколько сообщений из нескольких чатов одновременно, значит он может вести диалог между двумя пользователями, перенаправляя сообщения через себя, чем-то напоминая CRM-систему.

CRM-система – программное обеспечение, которое налаживает взаимодействие сотрудников организации и клиентов. Такие системы используют в компаниях, имеющих клиентскую поддержку. С развитием сервисов растет и нужда в поддержке клиентов, которые только начинают пользоваться сервисом, либо, которые хотят уточнить какой-либо вопрос. Для этой задачи требуются сотрудники, знающие, что нужно ответить на любой вопрос клиента. Учитывая множество факторов в ответе, новых сотрудников службы поддержки нужно обучать.

В не очень развитых организациях используют не эффективные системы обучения, что приводит к снижению работоспособности всех сотрудников. В компании, для которой будет создан бот, ранее использовался общий Telegram чат, в котором было более 100 участников. Десять из этих участников, являются кураторами групп, недавно вышедших из обучения, а остальные – группы стажеров. Как уже стало понятно, возникала сложность в обработке вопросов от новичков. Вопросы терялись среди всех сообщений, что приводило путанице для кураторов. В последствие, не удавалось подготовить отчетность для корректировки программы обучения. Было решено, что нужно сделать бота в Telegram (так как данный мессенджер был у всех сотрудников), способного упорядочить и распределить обращения от стажеров к кураторам. Таким образом, бот образует CRM-систему, которая налаживает взаимодействие кураторов и стажеров.

Прежде чем приступить к разработке, необходимо разобраться в том, что должен уметь бот.

Во-первых, боту нужно отличить куратора от стажера и один отдел от другого, из чего следует, что должна вестись запись пользователей, отделов, в которых они состоят, и выдача им ролей. Будем считать, что любой пользователь, который воспользовался ботом в первый раз является стажером. В таком случае, необходима роль администратора, способного любого пользователя сделать куратором. Также стоит учесть, что стажеры после определенного времени должны будут завершить свою практику, поэтому добавим статус заблокированного пользователя, не имеющего возможности взаимодействовать с ботом.

Во-вторых, чтобы не возникало путаницы в обращениях, следует сделать очередь и правильное распределение вопросов.

В-третьих, главная задача CRM-системы – это сбор статистики для улучшения качества работы, – поэтому будем вести историю сообщений пользователей.

Так как необходимо хранить много информации и структурировать ее, начнем с проектирования базы данных. В качестве СУБД будет использован PostgreSQL. Схема базы данных представлена на рисунке 1.

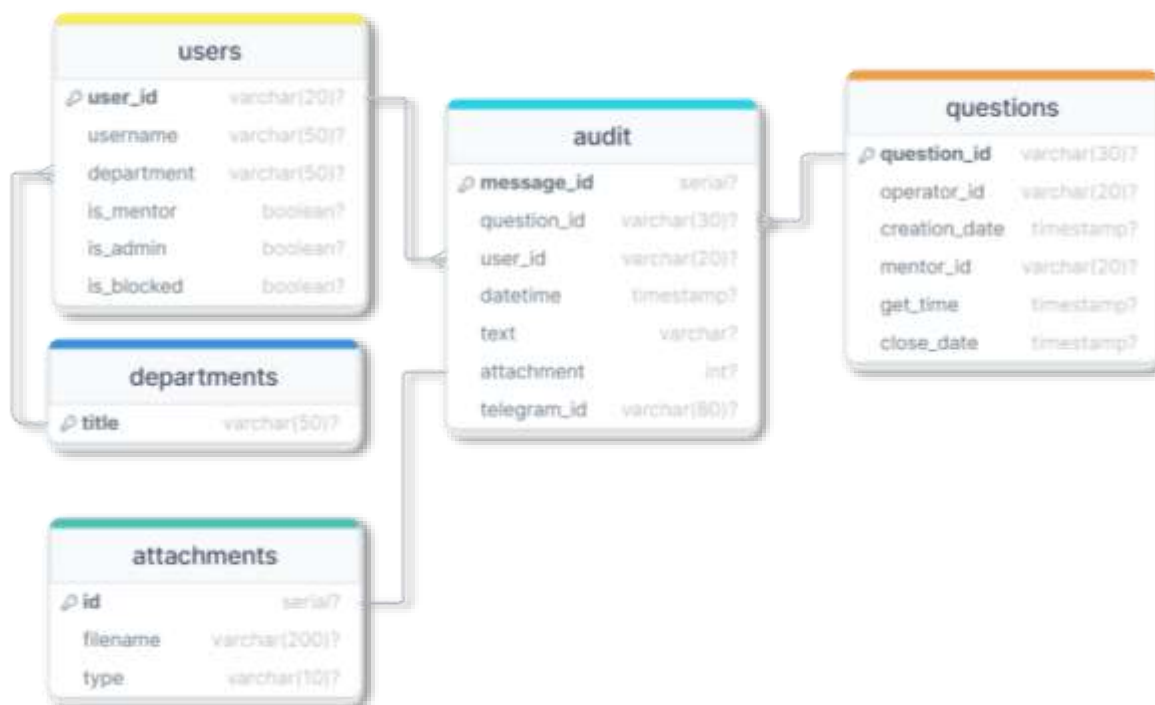


Рисунок 1 – Схема базы данных для бота

Теперь, когда база данных готова, можно приступить к разработке бота относительно задач, используя библиотеку `ruTelegramBotAPI` на языке программирования Python.

Роль пользователя определяется в хендлерах (функция, которая выполняет какие-либо действия в зависимости от содержания сообщения), пример изображен на рисунке 2. В функцию передается сообщение от пользователя и список всех пользователей.

```
bot.register_message_handler(send_ask_command,  
                             func=lambda m: user_is_default(m, bot.Users)  
                             content_types=['text'],  
                             pass_bot=True)
```

Рисунок 2 – Определение роли пользователя в хендлере

Очередь в боте организована обычным списком и взаимодействием с ним (рисунок 3).

```
def add_to_questions(self, user_id: int):  
    if self.any_mentor_online():  
        question = Question(user_id, self.department)  
        self.questions.append(question)  
        return self.add_mentor_for_question(question)  
    return -2
```

Рисунок 3 – Организация очереди в виде списка

Распределение вопросов сильно зависит от некоторых условий. Сначала проверяется наличие куратора. В случае его присутствия, возвращается код пользователя этого куратора. Далее проверяется первенство вопроса. Если данный вопрос и первый из очереди совпадают, то ищется куратор, который на данный момент не занимается никаким вопросом. Если куратор найден, то его прикрепляют к вопросу и обновляют списки вопросов и наставников. В противных случаях возвращают None, что означает «Ничего». Реализация распределения вопросов представлена на рисунке 4.

```
def add_mentor_for_question(self, question: Question):  
    if question.mentor_id is not None:  
        return question.mentor_id  
  
    if self.questions[0] is question:  
        mentor = self.find_free_mentor()  
  
        if mentor is not None:  
            self.mentors.remove(mentor)  
            self.questions.remove(question)  
  
            question.attach_mentor(mentor)  
  
            self.line.append(question)  
            self.mentors.append(mentor)  
  
        return mentor  
  
    return None
```

Рисунок 4 – Распределение вопросов от стажёров для кураторов

Запись истории сообщений ведется через проверку вложений к сообщениям. Также вместе с сообщением можно передать код вопроса от стажёра, что помогает сохранять полные диалоги пользователей, связанные с вопросом. Реализация этой функции представлена на рисунке 5.

```
def write_audit(message: Message, question_id='0'):
    att_id = 0
    if message.content_type == 'photo':
        att_id = db.add_attachment(message.photo[0].file_id, 'photo')

    if message.content_type == 'document':
        att_id = db.add_attachment(message.document.file_id, 'document')

    if message.content_type == 'sticker':
        att_id = db.add_attachment(message.sticker.file_id, 'sticker')

    db.add_audit(message, att_id, question_id)
```

Рисунок 5 – Запись истории сообщений в боте

Итак, бот в Telegram способен не только на отправку заготовленного ответа на сообщения пользователей, но и на распределение и перенаправление сообщений, что позволяет делать крупные системы. Благодаря BotAPI мессенджера, возможно сделать CRM-систему, разработка которой была представлена в этой работе. В ходе этой работы, было выполнено определение задач бота, проектирование базы данных и разработка программной части системы.

Список литературы

1. CRM-системы: что это такое, возможности, функции, задачи. Какие бывают и как их выбирать. – Текст: электронный // Блог Яндекс Практикума [сайт]. – 2021. – URL: <https://practicum.yandex.ru/blog/что-такое-crm-sistema-i-kak-ih-vybirat> (дата обращения: 13.10.2024).
2. Сидоров, В. А. CRM-системы в повышении эффективности сферы обращения / В. А. Сидоров, В. С. Казусь // Институционально-воспроизводственные механизмы российской экономики в условиях санкций : Материалы по итогам российской научной конференции, Краснодар, 14 мая 2019 года. – Краснодар: Научно-исследовательский институт экономики Южного федерального округа, 2019. – С. 143-151. – EDN RQVTVC.
3. Руководство по языку программирования Python. – Текст: электронный // METANIT.COM – Сайт о программировании [сайт]. – 2024. – URL: <https://metanit.com/python/tutorial/> (дата обращения 15.10.2024).

4. PyTelegramBotAPI: Документация к библиотеке PyTelegramBotAPI: [сайт]. – 2024. – URL: <https://pytba.readthedocs.io/ru/latest/index.html> (дата обращения: 16.10.2024). – Текст: электронный.

5. Руководство по PostgreSQL. – Текст: электронный // METANIT.COM – Сайт о программировании [сайт]. – 2022. – URL: <https://metanit.com/sql/postgresql/> (дата обращения 13.10.2024).

6. Зольников, К. В. Моделирование и оптимизация конструкции полосового фильтра на основе коаксиального резонатора / К. В. Зольников, Д. А. Ачкасов // Моделирование систем и процессов. – 2024. – Т. 17, № 2. – С. 43-50. – DOI 10.12737/2219-0767-2024-17-2-43-50. – EDN POHLTW.

References

1. CRM systems: what it is, features, functions, tasks. What are the types and how to choose them. – Text: electronic // Yandex Practicum blog [website]. – 2021. – URL: <https://practicum.yandex.ru/blog/chto-takoe-crm-sistema-i-kak-ih-vybirat> (date of request: 13.10.2024).

2. Sidorov, V. A. CRM systems in improving the efficiency of the sphere of circulation / V. A. Sidorov, V. S. Kazus // Institutional and reproductive mechanisms of the Russian economy under sanctions: Materials on the results of the Russian scientific conference, Krasnodar, May 14, 2019. – Krasnodar: Scientific Research Institute of Economics of the Southern Federal District, 2019. – pp. 143-151. – EDN RQVTVC.

3. Guide to the Python programming language. – Text: electronic // METANIT.COM – Website about programming [website]. – 2024. – URL: <https://metanit.com/python/tutorial/> (accessed 10/15/2024).

4. PyTelegramBotAPI: Documentation for the PyTelegramBotAPI library: [website]. – 2024. – URL: <https://pytba.readthedocs.io/ru/latest/index.html> (date of publication: 16.10.2024). – Text: electronic.

5. PostgreSQL Manual. – Text: electronic // METANIT.COM – Website about programming [website]. – 2022. – URL: <https://metanit.com/sql/postgresql/> (accessed 13.10.2024).

6. Zolnikov, K. V. Modelling and optimization of a bandpass filter design on the basis of a coaxial resonator / K. V. Zolnikov, D. A. Achkasov // Modelling of systems and processes. – 2024. – Т. 17, № 2. – С. 43-50. – DOI 10.12737/2219-0767-2024-17-2-2-43-50. – EDN POHLTW.