

Разработка и оптимизация бота в Telegram для эффективного управления задачами на предприятии

В.А. Васькин, С.Д. Шибайкин, В.В. Никулин

Национальный исследовательский Мордовский государственный университет
им. Н.П. Огарева,
430005, г. Саранск, ул. Большевистская, д. 68, Россия

Резюме. Цель. Целью исследования является разработка и реализации Telegram-бота для эффективного управления задачами с использованием функциональных возможностей Telegram-API. **Метод.** Исследование основано на методах проектирования базы данных. **Результат.** В работе описан процесс разработки и оптимизации Telegram-бота для эффективного управления задачами. В рамках исследования было разработано приложение, реализованное на языке Python с применением Telegram API и использующее СУБД PostgreSQL для хранения данных в рамках рабочего процесса. Система предназначена для оптимизации рабочего процесса, управления задачами в системе и делегирования их пользователям. Спроектирована база данных, где определены структура и связи между данными, обеспечивающие эффективное хранение и доступ к информации с учетом потребностей приложения. **Вывод.** Поэтапное внедрение функциональности, от управления задачами, системы оповещения пользователей, до обработки и анализа данных, позволяет получить полнофункциональную систему с широким спектром возможностей. Для прояснения функций и возможностей бота проведена демонстрация работы приложения, отражающая функциональные возможности системы.

Ключевые слова: базы данных, бот, приложение, проектирование, оптимизация, управление

Для цитирования: В.А. Васькин, С. Д. Шибайкин, В.В. Никулин. Разработка и оптимизация бота в Telegram для эффективного управления задачами на предприятии. Вестник Дагестанского государственного технического университета. Технические науки. 2024; 51(2):70-76. DOI:10.21822/2073-6185-2024-51-2-70-76

Development and optimization of a bot in Telegram for effective task management in the enterprise

V.A. Vaskin, S.D. Shibaikin, V.V. Nikulin

N.P. Ogarev National Research Mordovia State University,
68 Bolshevistskaya St., Saransk 430005, Russia

Abstract. Objective. The purpose of the research is to develop and implement a Telegram bot for effective task management using the functionality of the Telegram API. **Method.** The study is based on database design techniques. **Result.** The paper describes the process of developing and optimizing a Telegram bot for effective task management. As part of the research, an application was developed, implemented in Python using the Telegram API and using the PostgreSQL DBMS to store data as part of the workflow. The system is designed to optimize the workflow, manage tasks in the system and delegate them to users. A database has been designed, where the structure and relationships between data are defined, ensuring efficient storage and access to information, taking into account the needs of the application. **Conclusion.** The phased implementation of functionality, from task management, user notification systems, to data processing and analysis, allows us to obtain a fully functional system with a wide range of capabilities.

ties. To clarify the functions and capabilities of the bot, a demonstration of the application was carried out, reflecting the functionality of the system.

Ключевые слова: databases, bot, application, design, optimization, management

For citation: V.A. Vaskin, S.D. Shibaikin, V.V. Nikulin. Development and optimization of a bot in Telegram for effective task management in the enterprise. Herald of Daghestan State Technical University. Technical Sciences. 2024; 51(2):70-76. DOI:10.21822/2073-6185-2024-51-2-70-76

Введение. Современные бизнес-процессы требуют от предприятий эффективного управления задачами и проектами, что важно не только с точки зрения обеспечения чёткости и структурированности задач на этапе их составления, но и эффективности их распределением между сотрудниками. Разработка системы для управления задачами представляет собой не только средство для оперативного взаимодействия с задачами, но и инструмент для немедленного получения уведомлений, установки приоритетов и мониторинга текущего состояния задач.

Постановка задачи. Быстрота и удобство использования мессенджера Telegram, а также его широкий охват в сфере корпоративной коммуникации делают его идеальной платформой для интеграции системы управления задачами [1–3]. Пользователи получают возможность быстро реагировать на изменения, управлять приоритетами и взаимодействовать с системой без необходимости переключения между различными приложениями.

Методы исследования. Эффективное проектирование базы данных является фундаментальным шагом в разработке системы управления задачами. В качестве реляционной базы данных для хранения информации о пользователях, сообщениях и текущей загрузке ответственных лиц используется PostgreSQL [4].

На данном этапе разработки система имеет 3 таблицы (рис. 1). Таблица «users» предназначена для хранения информации о пользователях. Информация таблицы позволяют системе определять занимаемую пользователем позицию на предприятии, его рабочую нагрузку и использовать личные данные пользователя (ссылка на Telegram-аккаунт) для дальнейшего оповещения по новым задачам.

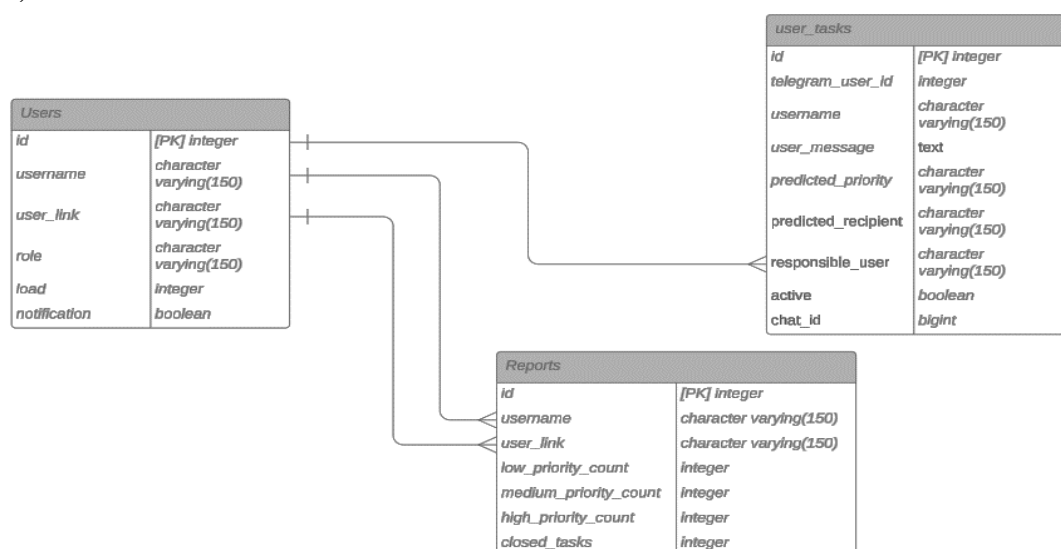


Рис. 1. Структура таблиц данных

Fig. 1. Structure of data tables

Таблица «user_tasks» предназначена для хранения информации о сообщениях, содержащих задачи, включая идентификатор сообщения, идентификатор пользователя, открывшего задачу, предсказанный приоритет посредством нейронной сети, а также ответственное лицо, которому задача была делегирована.

Таблица reports содержит информацию о пользователях и их активности в системе. Каждая запись представляет отдельного пользователя, указывая его идентификатор, имя пользователя, ссылку на профиль и статистику выполненных задач.

Взаимодействие бота с базой данных PostgreSQL достигается путем использования библиотеки «psycopg2» для языка Python [4]. Эта библиотека предоставляет возможность подключения к базе данных, выполнения SQL-запросов и получения результатов. Запросы включают в себя выборку информации о пользователях, вставку новых сообщений о задачах, обновление данных после назначения задачи и другие операции.

В качестве примера приводится SQL-запрос для обновления данных таблицы reports, способствующий формированию отчёта:

```
cur = conn.cursor()
cur.execute("""
    INSERT INTO combined_table (user_id, username, user_link, low_priority_count, medium_priority_count, high_priority_count, closed_tasks)
    SELECT
        u.id AS user_id,
        u.username,
        u.user_link,
        COUNT(CASE WHEN um.predicted_priority = 'Низкий' AND um.active = TRUE THEN 1
        ELSE NULL END) AS low_priority_count,
        COUNT(CASE WHEN um.predicted_priority = 'Средний' AND um.active = TRUE THEN 1
        ELSE NULL END) AS medium_priority_count,
        COUNT(CASE WHEN um.predicted_priority = 'Высокий' AND um.active = TRUE THEN 1
        ELSE NULL END) AS high_priority_count,
        COUNT(CASE WHEN um.active = FALSE THEN 1 ELSE NULL END) AS closed_tasks
    FROM
        users u
    LEFT JOIN
        user_messages um ON u.user_link = um.responsible_user
    GROUP BY
        u.id, u.username, u.user_link
conn.commit()
```

Запрос на обновление сводит вместе данные из таблицы «users» и «user_tasks», группируя их по пользователям. При помощи различных агрегатных функций рассчитывается количество задач каждого приоритета (низкого, среднего, высокого) и закрытых задач для каждого пользователя. В результате выполнения запроса обновления формирует отчёт, содержащий статистику пользователей по задачам.

Бот для эффективного управления задачами на предприятии обладает следующими функциональными возможностями:

- обработка пользовательских задач;
- управление задачами (возможность их просмотра и удаления);
- управление пользователями;
- система оповещений;
- система генерации отчетов;

Ключевым компонентом системы управления задачами является функция обработки входящих сообщений. Она запускается при получении нового сообщения от пользователя, что способствует выполнению следующих шагов.

Первым шагом в обработке сообщений является анализ текста вопроса пользователя и сбор информации о нём. Система использует предварительно обученные модели нейронной сети для автоматического предсказания приоритета и адресата задачи на основе текста вопроса пользователя [5, 6]. Прогнозируемые приоритеты и получатели служат важными исходными данными для классификации задач.

В зависимости от предсказанного приоритета, система определяет уровень загрузки задачами, который будет использоваться при распределении задач между ответственными лицами. Каждой из задач присваивается вес в зависимости от приоритета.

Система рассчитана на два сценария поведения. Первый предусматривает использование приложения сотрудниками IT-компаний с использованием интеграции с обученной моделью предсказания приоритетов. Если обученная модель не работает или не требуется для организации задач на предприятии – приложение поддерживает режим работы без нее. В этом случае уведомление о новой задаче приходит без приоритета выполнения. Используя язык SQL [7], бот обращается к базе данных PostgreSQL для выбора пользователя с минимальной нагрузкой, который будет ответственным за выполнение новой задачи. По завершению обработки сообщения ответственному пользователю приходит оповещение о новой задаче. Реализация представлена следующим фрагментом кода:

```
cur.execute("UPDATE users SET load = %s WHERE id = %s", (result_load))
cur.execute("SELECT MAX(id) AS last_id FROM user_messages")
response = f'Зарегистрирована задача номер {task_id}\nПользователь: {user_link}\nПриоритет:{predicted_priority}\nНаправление:{predicted_recipient}\nОтветственный: {user_link_response}'
context.bot.send_message(chat_id=update.message.chat_id, text=response)
cur.execute("SELECT chat_id from users WHERE user_link = %s", (user_link_response,))
chat_id = cur.fetchall()
response_for_user = f'Вам поступила новая задача. Номер задачи {task_id}\nПользователь:{user_link}\nПриоритет: {predicted_priority}\nНаправление: {predicted_recipient}'
bot.send_message(id_chat, response_for_user)
```

Эти шаги обеспечивают эффективное управление задачами, начиная с обработки входящих сообщений пользователей, распределением задачи между ответственными лицами, а также оповещения пользователей по новым задачам.

Отслеживание статистики является одним из ключевых аспектов работы приложения. Функция report автоматизирует создание отчетов, содержащих сводку по статистике задач для каждого пользователя. Информация структурируется в виде Excel-файла и отправляется в чат Telegram, из которого была вызвана функция.

Основная цель функции – предоставить пользователям всесторонний обзор распределения и выполнения задач в команде. Путем использования базы данных PostgreSQL и библиотеки Pandas, функция динамически запрашивает данные, обрабатывает их и формирует Excel-отчет. Этот отчет впоследствии отправляется в чат Telegram.

Функция включает в себя следующие шаги.

- Установление соединения с базой данных PostgreSQL.
- Получение обновленных данных из таблицы reports.
- Генерация Excel-отчета на основе DataFrame.
- Отправка Excel-файла в чат Telegram в качестве документа.

Интеграция запросов к базе данных, обработки данных и создания документов демонстрирует эффективность и универсальность данного подхода.

Обсуждение результатов. Далее рассмотрим ключевые моменты взаимодействия пользователя с приложением, начиная от отправки запроса и заканчивая получением результата. Демонстрация отражает использования функций бота, а также обеспечивает полное представление о том, как приложение может быть внедрено в бизнес-процессы.

Для того, чтобы запустить процесс обработки сообщения ботом, следует написать в чат с указанием имени бота через @. Режим работы, при котором у бота нет доступа к обученной модели для генерации приоритетов задач сотрудников IT-компаний, не отличается от представленных на рис. 2 результатов.

Сообщение, отправленное ботом, сигнализирует о том, что обработка сообщения прошла успешно, ботом были собраны данные о пользователе, который задачу отправил,

нейронная сеть определила приоритет задачи, её область, а система распределения пользователей назначила ответственного. В этот момент сотруднику предприятия приходит оповещение в чате самим ботом о том, что для него была зарегистрирована конкретная задача, под определённым номером.

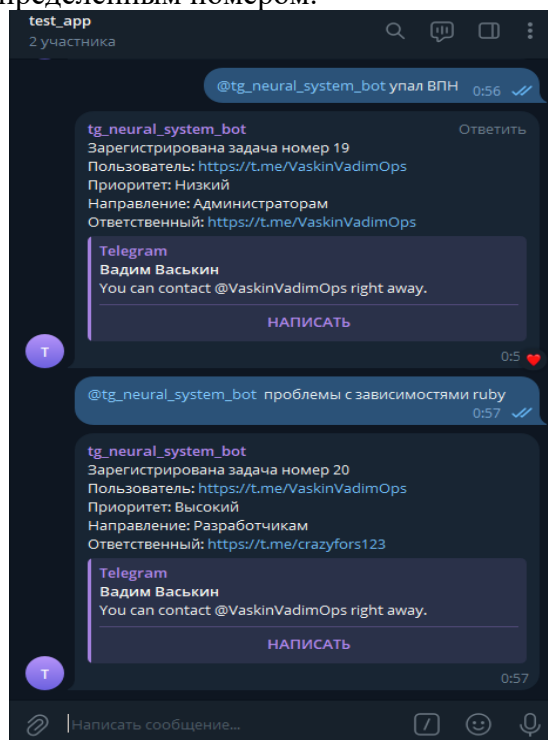


Рис. 2. Окно чата с ботом
Fig. 2. Chat window with bot

Администратор может посмотреть список доступных ему пользователей посредством команды /showusers (рис.3). Бот предоставит ему список из Inline кнопок, каждую из которых можно выбрать для просмотра задач, делегированных пользователю.

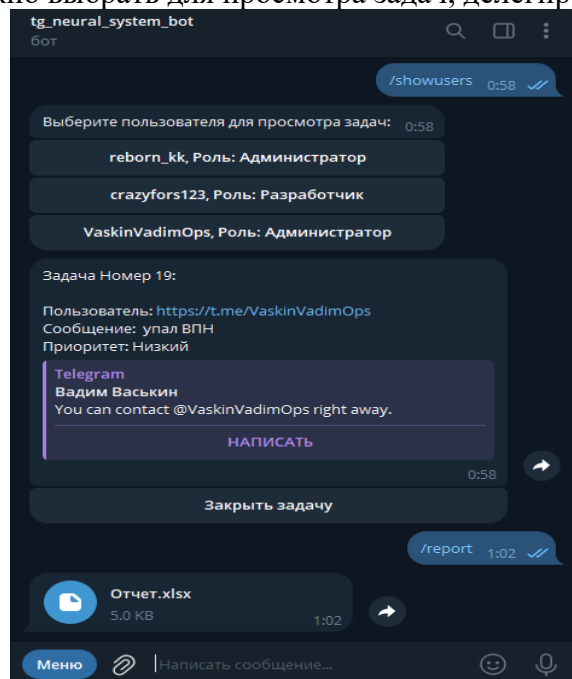


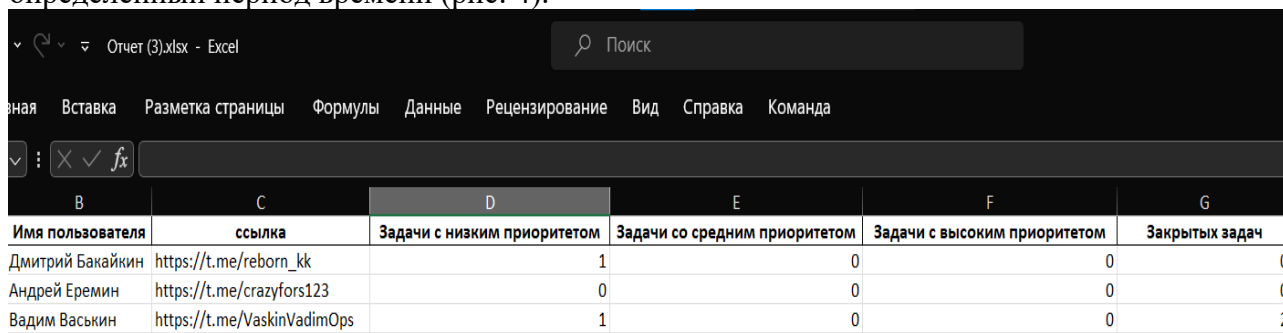
Рис. 3. Окно выбора задач для просмотра
Fig. 3. Window for selecting tasks to view

Представленный выше результат иллюстрирует панель администратора приложения. На деле, каждый из сотрудников, занимающихся задачами, видит исключительно

задачи, делегированные ему. Сотрудник имеет возможность закрыть задачу по факту её выполнения или неактуальности. После нажатия на кнопку Inline, поступит запрос в базу данных, задача будет переведена в статус закрытой, а пользователю в чат придёт оповещение, что задача была закрыта администратором.

Для актуализации данных по задачам и статуса их выполнения бот предусматривает возможность генерации отчетов по проделанной работе.

В результате выполнения SQL-запроса формируется Excel-файл, в котором представлена статистика пользователей – информация о нагрузке пользователей задачами за определенный период времени (рис. 4).



Имя пользователя	ссылка	Задачи с низким приоритетом	Задачи со средним приоритетом	Задачи с высоким приоритетом	Закрытых задач
Дмитрий Бакайкин	https://t.me/reborn_kk	1	0	0	0
Андрей Еремин	https://t.me/crazyfors123	0	0	0	0
Вадим Васюкин	https://t.me/VaskinVadimOps	1	0	0	2

Рис. 4. Вид сформированной таблицы статистики пользователей

Fig. 4. View of the generated user statistics table

Возможность автоматического формирования отчетов и их отправки в удобном для пользователя формате дополняет функциональность системы, делая ее более привлекательной для бизнес-пользователей.

Вывод. В статье был представлен процесс разработки и реализации Telegram-бота для эффективного управления задачами с использованием функциональных возможностей Telegram-API, рассмотрены ключевые этапы разработки системы, начиная с проектирования базы данных, написание API и формирования функциональных возможностей системы. Результаты исследования демонстрируют высокую эффективность системы.

Важно подчеркнуть, что разработанная система имеет потенциал для широкого применения в разных отраслях, где управление задачами играет ключевую роль, а также гибкость разработанного приложения внедряться в различные сферы деятельности, включая IT, производство, маркетинг и многие другие.

Ее способность адаптироваться под уникальные потребности и задачи компаний делает ее перспективным решением для организаций, стремящихся оптимизировать управление задачами, улучшить производительность и повысить общую конкурентоспособность.

Библиографический список:

1. Абрамова, А.И. Возможности использования мессенджера Telegram в сфере высшего образования / А.И. Абрамова // Студенческая наука: актуальные вопросы, достижения и инновации : Сборник статей VIII Международной научно-практической конференции, Пенза, 07 июня 2022 года. – Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.), 2022. – С. 53-55.
2. Бийбосунов Б.И., Бийбосунова С.К., Жолочубеков Н.Ж. Описание концепции Telegram ботов и их разработка // Colloquium-Journal. 2020. № 7-1 (59). С. 7-11.
3. Омельченко Д.А., Гонатаев Р.Г., Фешина Е.В., Куштанок С.А. Разработка Telegram бота на языке программирования Python // Наука XXI века: проблемы, перспективы и актуальные вопросы развития общества. Международная межвузовская весенняя научно-практическая конференция. Яблоновский, 2021. С. 109-111.
4. Введение в PostgreSQL с Python +Psycopg2 [Электронный ресурс]/ – URL: <https://pythonru.com/biblioteki/vvedenie-v-postgresql-s-python-psycopg2/> (дата обращения: 01.02.2024).
5. Никулин В.В., Шибайкин С.Д., Соколова М.С. Применение методов машинного обучения для автоматизированной классификации и маршрутизации в библиотеке ITIL // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2022. № 1. С. 42–52.

6. Shibaikin S.D., Nikulin V.V., Vishnyakov A.N. Application of Machine Learning Methods for Automated Classification and Routing in ITIL. // Journal of Physics: Conference Series. 5. Ser. «5th International Scientific Conference on Information, Control, and Communication Technologies, ICCT 2021» 2021. C. 012041.
7. Гринченко, Н.Н., Хизриева Н.И. Базы данных. Программирование на SQL. – Москва : Общество с ограниченной ответственностью Издательство «КУРС», 2023. – 240 с.

References:

1. Abramova, A.I. Possibilities of using the Telegram messenger in the field of higher education / A.I. Abramova. Student science: current issues, achievements and innovations: Collection of articles of the VIII International Scientific and Practical Conference, Penza, June 07, 2022. – Penza: Science and Enlightenment (IP Gulyaev G.Yu.), 2022; 53-55. (In Russ)
2. Biybosunov B.I., Biybosunova S.K., Zholochubekov N.Zh. Description of the concept of Telegram bots and their development. *Colloquium-Journal*. 2020;7-1 (59): 7-11. (In Russ)
3. Omelchenko D.A., Gonataev R.G., Feshina E.V., Kushtanok S.A. Development of a Telegram bot in the Python programming language. Science of the XXI century: problems, prospects and current issues of social development. International interuniversity spring scientific and practical conference. Yablonovsky, 2021; 109-111. (In Russ)
4. Introduction to PostgreSQL with Python +Psycopg2 [Electronic resource]/ – URL: <https://pythonru.com/biblioteki/vvedenie-v-postgresql-s-python-psycopg2> (access date: 02/01/2024). (In Russ)
5. Nikulin V.V., Shibaykin S.D., Sokolova M.S. Application of machine learning methods for automated classification and routing in the ITIL library. *Bulletin of the Astrakhan State Technical University. Series: Management, computer technology and information science*. 2022;1: 42–52. (In Russ)
6. Shibaikin S.D., Nikulin V.V., Vishnyakov A.N. Application of Machine Learning Methods for Automated Classification and Routing in ITIL. *Journal of Physics: Conference Series. 5. Ser. "5th International Scientific Conference on Information, Control, and Communication Technologies, ICCT 2021"* 2021; 012041. (In Russ)
7. Grinchenko, N.N., Khizrieva N.I. Databases. SQL programming. Moscow: Limited Liability Company Publishing House “KURS”, 2023; 240.

Сведения об авторах:

Васькин Вадим Александрович, магистрант 2 года обучения, кафедра автоматизированных систем обработки информации и управления; vaskin2000mordvinrus@mail.ru

Шибайкин Сергей Дмитриевич, кандидат технических наук, доцент, доцент, кафедра инфокоммуникационных технологий и систем связи; shibaikinsd@mail.ru

Никулин Владимир Валерьевич, кандидат технических наук, доцент, доцент, заведующий кафедрой инфокоммуникационных технологий и систем связи; nikulinvv@mail.ru

Information about authors:

Vadim .A. Vaskin, Master's student 2 years of study at the Department of Automated Information Processing and Control Systems; vaskin2000mordvinrus@mail.ru

Sergei D. Shibaikin, Cand. Sci. (Eng.), Assoc. Prof., Assoc. Prof., Department of Infocommunication Technologies and Communication Systems; shibaikinsd@mail.ru

Vladimir.V. Nikulin, Cand. Sci. (Eng.), Assoc. Prof., Head of the Department of Infocommunication Technologies and Communication Systems; nikulinvv@mail.ru

Конфликт интересов/Conflict of interest.

Авторы заявляют об отсутствии конфликта интересов/The authors declare no conflict of interest.

Поступила в редакцию/Received 07.02.2024.

Одобрена после рецензирования/ Reviced 27.02.2024.

Принята в печать/Accepted for publication 27.02.2024.