***К.Ю. Кожевников, В.С. Иванов, Э.Е. Ярдыков,   
ст. преп. Н.В. Первова***

**ТЕЛЕГРАМ-БОТ ДЛЯ БРОНИРОВАНИЯ МЕСТ В  
КОВОРКИНГЕ**

**Аннотация.** Описывается концепция работы телеграм-бота для бронирования мест в коворкинге.

**Ключевые слова:** телеграм-бот, телеграм, Python, MySQL, коворкинг.

***K.Y. Kozhevnikov, V.S. Ivanov, E.E. Yardykov,   
Senior Lecturer N.V.Pervova***

**TELEGRAM-BOT FOR COWORKING SPACE  
RESERVATION**

**Annotation.** Describes the concept of a telegram-bot for booking places in a co-working space.

**Keywords:** telegram-bot, Telegram, Python, MySQL, coworking.

В современном мире концепция коворкинга становится все более популярной, особенно среди студентов и молодых специалистов, стремящихся к продуктивной и вдохновляющей рабочей среде. Коворкинг представляет собой совместное использование рабочего пространства, где люди с различными профессиями и областями интересов могут собираться в одном месте для работы, обмена идеями и совместного развития.

Проект телеграм-бота, разработанный для совместной работы с веб-платформой бронирования мест коворкинга, является важным инструментом в реализации этой концепции. Благодаря боту, студенты получают возможность быстро бронировать места в коворкинге, не нуждаясь в запуске сайта на телефонах, так как он является более удобным в случае использования небольших экранов.

Для создания телеграм-бота для бронирования рабочих мест в университетском коворкинге используются современные технологии и инструменты. Основой разработки является язык программирования Python, который обеспечивает простоту и гибкость использования. pyTelegramBotAPI и pyMySQL обеспечивают удобство и надёжную работу телеграм бота и его базы данных. Для хранения данных о бронированиях и другой информации используется база данных MySQL, обеспечивающая эффективное хранение и быстрый доступ к данным. В качестве редактора кода используется Visual Studio Code, который обеспечивает удобную среду разработки.

Функционал телеграм бота включает в себя выбор конкретного места в коворкинге, а также удобную настройку времени бронирования. Дополнительно, для более эффективной организации рабочего пространства, предусмотрен а возможность указания количества человек, которые планируют занять место (по количеству уникальных номеров студенческих билетов). Это позволяет оптимизировать использование ресурсов и предоставить возможность работать в коворкинге не только индивидуально, но и в команде.

Реализация телеграм бота с использованием pyTelegramBotAPI начинается с создания бота у @BotFather, получения токена и конфигурации данных.

Для вывода приветственного сообщения после нажатия кнопки /start (рис.1), задействуется следующий код:

@bot.message\_handler(commands=['start', 'help'])

Def send\_welcome(message):

bot.reply\_to(message,f"Привет,{message.from\_user.username}\nИспользуй команду '/reserve', чтобы записаться")

sql.create\_new\_user\_in\_table(message.from\_user.id)

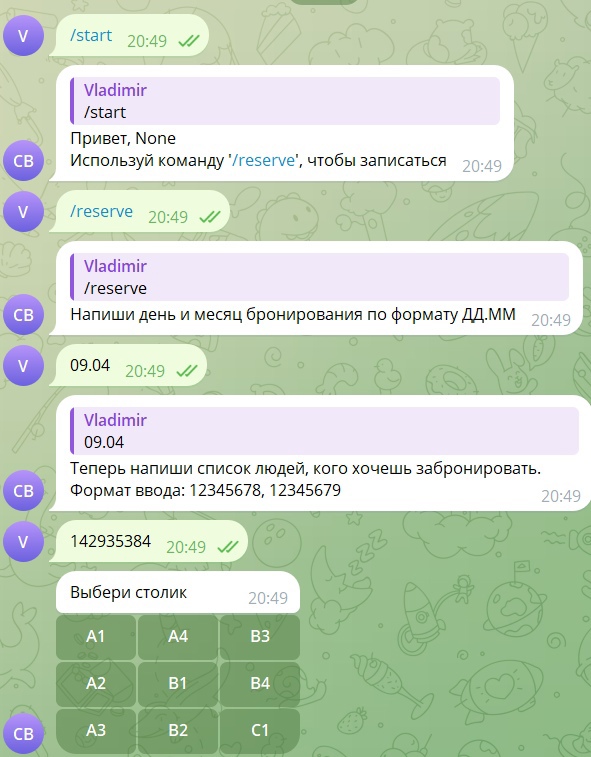


Рис. 1. Вывод приветствия бота

Для выбора месяца и дня бронирования (рис.2) используется уже другой код:

@bot.message\_handler(commands = ['reserve'])

Def \_reserve(message):

bot.reply\_to(message, "Напиши день и месяц

бронирования по формату ДД.ММ")

sql.set\_status\_of\_user(message.from\_user.id,1)

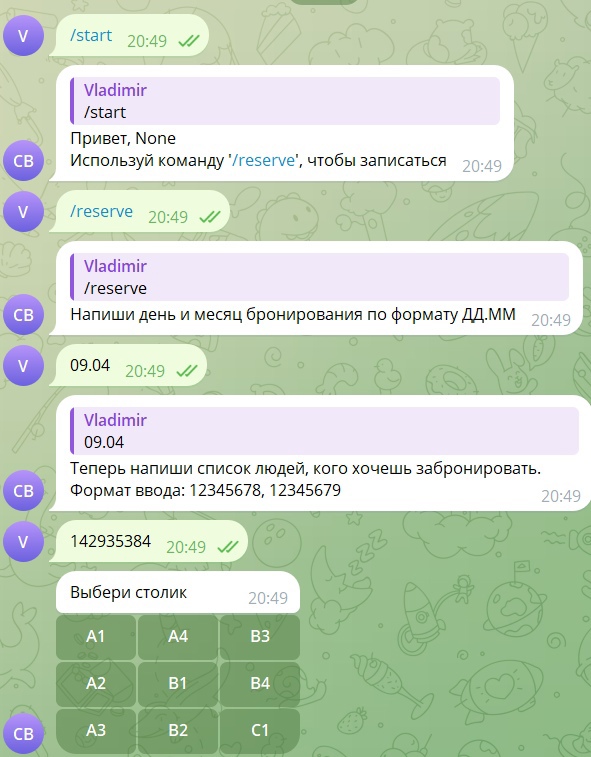


Рис. 2. Вывод дня и месяца для бронирования

После ввода даты пользователь указывает номера студенческих билетов записываемых участников (рис.3), реализация этого действия в коде выглядит так:

temp\_ = parse\_DATE\_and\_TIME(message.text, message)

if (temp\_ != False):

sql.user\_set\_data(f"{temp\_[0]},{temp\_[1]},

{datetime.now().year}", message.from\_user.id)

sql.set\_status\_of\_user(message.from\_user.id, 2)

bot.reply\_to(message, "Теперь напиши список

людей, кого хочешь забронировать. Формат

ввода: 12345678, 12345679")

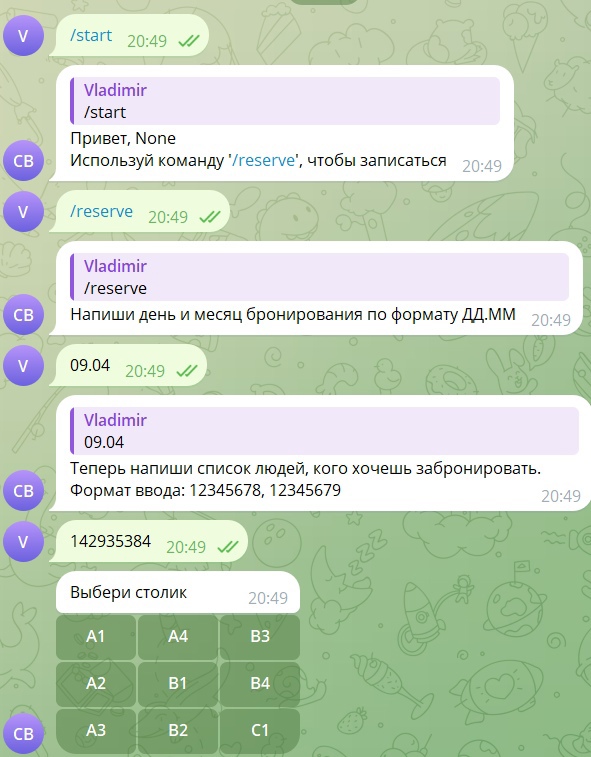


Рис. 3. Ввод данных участников

После этого пользователь может выбрать столик (рис.4):

temp\_ = \_\_check\_uids(message.text, message)

if (temp\_ != False):

sql.set\_status\_of\_user(message.from\_user.id, 3)

sql.user\_set\_uids(message.text,

message.from\_user.id)

bot.send\_message(message.chat.id,"Выбери столик",

reply\_markup=show\_slots)

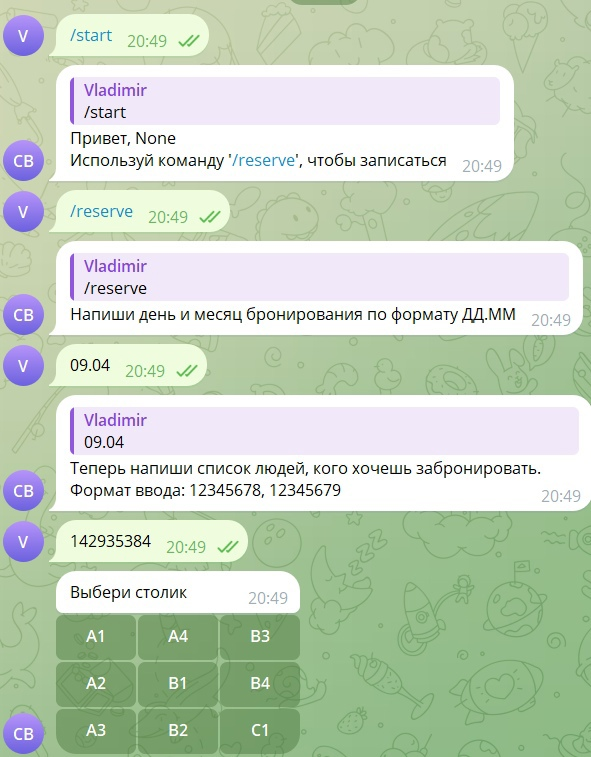


Рис. 4. Выбор рабочего места

При нажатии на выбранный столик отображается панель выбора временного интервала бронирования (рис.5):

if temp\_data\_for\_time[0][0]['text'][-1] != '❌':

sql.user\_set\_time(1, call.from\_user.id)

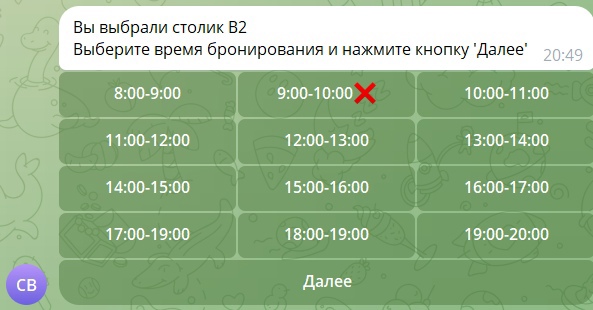


Рис.5 Выбор времени бронирования

После завершения бронирования бот выводит надпись (рис.6):

sql.send\_to\_main(call.from\_user.id)

sql.set\_status\_of\_user(call.from\_user.id, 5)

bot.delete\_message(call.message.chat.id, call.message.id)

bot.send\_message(call.message.chat.id, "Вы успешно записались!")

sql.clear\_user\_info(call.from\_user.id)



Рис. 6. Сообщение об успешной записи

Так же был создан дополнительный класс для работы с MySQL. Пример настройки подключения:

Connection = pymysql.connect(

host = HOST,

port = 3306,

user = USER,

password = PASS,

database = "coworking",

cursorclass = pymysql.cursors.DirtCursor,

autocommit = True

)

cursor = connection.cursor()

def \_\_init\_\_(self):

if(self.connection):

print("Подключились к бд")

else:

print("Ошибка подключения к бд")

Для каждого действия со стороны пользователя определены свои методы в классе для работы с MySQL. Так как бот расчитан на постоянную работу с большим количеством людей, которые, возможно, будут работать с ним одновременно, была использована дополнительная небольшая база данных, которая вмещает в себя служебные данные о пользователе. Они будут хранится там до тех пор, пока пользователь не завершит процесс бронирования места в коворкинге. После этого, служебные данные удалятся, освобождая память от уже устаревших данных.

Телеграм-бот играет важную роль в функционировании платформы коворкинга. Она обеспечивает простой и быстрый доступ к бронированию рабочих мест, что позволяет пользователям удобно и эффективно пользоваться платформой.

Литература

1. Документация pyTelegramBotAPI [Электронный ресурс]. Режим доступа: https://pytba.readthedocs.io/ru/latest/index.html (дата обращения: 15.03.2024).
2. Документация по PyMySQL [Электронный доступ]. Режим доступа: https://pymysql.readthedocs.io/en/latest/index.html (дата обращения: 15.03.2024).