Московский государственный технический университет им. Н.Э. Баумана.

Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»
Отчет по лабораторной работе №6.1
На тему «Разработка бота на основе конечного автомата для Telegram с использованием языка Python»

Выполнил:
студент группы ИУ5-31Б
Мишакин А.О.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2024

Цель работы: изучение разработки ботов в Telegram.

Задание: Разработайте бота для Telegram.

Описание: Идея данного бота заключается в том, чтобы он помогал с учёбой и повседневными делами. Данный бот умеет записывать твои дела и потом отправлять уведомление в момент их начала. Также у него можно попросить задать тебе квиз и дать варианты ответа (пока это просто вопрос в коде, но это можно расширить в двух направлениях. 1. Чтобы пользователь сам вписывал себе вопросы на будущее чтобы потом бот в случайном порядке выдавал их и предлагал варианты ответа. 2. Чтобы вопросы генерировались сами на заданную тему от пользователя, с помощью какого-то сайта или платформы.) Дополнительно проведена работа с кнопками для более простого пользования данным продуктом. Отчёт по данному боту разбит на 2 отчёта по просьбе преподавателя.

Текст программы:

```
import telebot
from telebot.types import ReplyKeyboardMarkup, KeyboardButton
import schedule
import threading
import time
import random

bot = telebot.TeleBot("7697829782:AAG9Y5-3_pnp6TurRY5A5FlNKC0ZAVThakU")
user_tasks = {}


@bot.message_handler(commands=['start'])
def start_bot(message):
    markup = ReplyKeyboardMarkup(resize_keyboard=True)
    btn_add_task = KeyboardButton("1. Добавить задачу")
    btn_show_tasks = KeyboardButton("2. Показать задачи")
    btn_start_quiz = KeyboardButton("3. Начать квиз")
    btn_help = KeyboardButton("4. Справка")
    markup.add(btn_add_task, btn_show_tasks)
    markup.add(btn_start_quiz, btn_help)

    bot.send_message(
        message.chat.id,
        "Привет! Выберите действие:",
        reply_markup=markup
    )

@bot.message_handler(func=lambda message: True)
def handle_buttons(message):
    if message.text == "1. Добавить задачу":
        add_task(message)
    elif message.text == "2. Показать задачи":
        show_tasks(message)
    elif message.text == "3. Начать квиз":
        start_quiz(message)
    elif message.text == "4. Справка":
        auxiliary_help(message)
    else:
        bot.send_message(message.chat.id, "Я не понял эту команду. Попробуйте
```

```
ещё раз.")

def add_task(message):
    bot.send_message(message.chat.id,"Введите задачу в формате: <тема>;<время
(чч:мм)>")
    bot.register_next_step_handler(message, save_task)

def save_task(message):
    try:
        topic, task_time = message.text.split(";")
        topic = topic.strip()
        task_time = task_time.strip()
        chat_id = message.chat.id

        if chat_id not in user_tasks:
            user_tasks[chat_id] = []
        user_tasks[chat_id].append((topic, task_time))
        schedule.every().day.at(task_time).do(send_notification, chat_id,
topic)
        bot.send_message(chat_id, f"Задача '{topic}' добавлена на
{task_time}." , reply_markup=start_keyboard())

    except Exception as e:
        bot.send_message(message.chat.id, "Ошибка! Проверьте формат задачи.
Пример: Прочитать книгу;15:30")
```