

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

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по домашнему заданию

«Бот: кулинарный помощник»

Выполнил:

студент группы ИУ5-31Б

Сомов Кирилл

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2023 г.

Описание проекта:

Разработка Telegram-бота, который будет предоставлять пользователю кулинарные рецепты на основе выбранных им ингредиентов.

Функциональные требования:

/start:

Приветственное сообщение при старте бота.

/help:

Предоставление справочной информации о возможностях бота.

/ingredients:

Вывод списка категорий продуктов для выбора.

При выборе категории - вывод доступных ингредиентов из этой категории.

Добавление ингредиентов:

После выбора ингредиента пользователем, он добавляется в список.

/view_ingredients:

Просмотр текущего списка добавленных ингредиентов.

/remove_ingredient:

Удаление последнего добавленного ингредиента из списка.

/clear:

Очистка списка добавленных ингредиентов.

/cook:

Поиск подходящего рецепта на основе списка ингредиентов пользователя.

Вывод наилучшего рецепта и списка недостающих ингредиентов, если такие есть.

Требования к базе данных:

Использование JSON-файлов для хранения данных о категориях ингредиентов и рецептах.

Логика работы бота:

При выборе категории или ингредиента - формирование клавиатуры с вариантами выбора.

Проверка наличия ингредиента в списке пользователя перед добавлением.

Поиск рецепта осуществляется на основе пересечения ингредиентов пользователя с рецептами.

Обработка ошибок и неопределенных команд:

В случае неопределенной команды выводить сообщение об ошибке и предложение воспользоваться /help для получения справки.

Технологии:

Использование библиотеки Telebot для работы с Telegram API.

Язык программирования: Python.

Код программы

```
import telebot
import json
from collections import Counter

TOKEN = '6325303584:AAEx2PeNZBpPnmdfXTFzNyXna-VyxbQNfzg'
bot = telebot.TeleBot(TOKEN)

# Загрузка данных о ингредиентах из файла
with open('ingredients.json', 'r', encoding='utf-8') as file:
    ingredients_data = json.load(file)

# Загрузка данных о рецептах из файла
with open('recipes.json', 'r', encoding='utf-8') as file:
    recipes_data = json.load(file)

user_ingredients = []

# Обработчик команды /start
@bot.message_handler(commands=['start'])
def handle_start(message):
```

```

bot.send_message(message.chat.id,
                  "Привет! Я кулинарный помощник. Давай выберем блюдо! Напиши /help,
чтобы узнать больше.")

# Обработчик команды /help
@bot.message_handler(commands=['help'])
def handle_help(message):
    help_message = (
        "Привет! Я кулинарный помощник.\n"
        "Для начала напиши /ingredients и выбери категорию продуктов.\n"
        "Затем добавляй продукты в свой список. Когда закончишь, напиши /cook, чтобы я
нашел подходящий рецепт.\n"
        "Если хочешь посмотреть эту справку, напиши /help.\n\n"
        "Дополнительные команды:\n"
        "/view_ingredients - посмотреть текущие ингредиенты в списке\n"
        "/remove_ingredient - удалить последний добавленный ингредиент из списка\n"
    )
    bot.send_message(message.chat.id, help_message)

# Обработчик команды /ingredients
@bot.message_handler(commands=['ingredients'])
def handle_ingredients(message):
    markup = create_categories_markup()

    bot.send_message(message.chat.id, "Выбери категорию продуктов:",
reply_markup=markup)

# Обработчик выбора категории продуктов
@bot.message_handler(func=lambda message: message.text in ingredients_data)
def handle_category(message):
    category = message.text

    # Запрос ингредиентов из выбранной категории
    markup = telebot.types.ReplyKeyboardMarkup(resize_keyboard=True)
    for ingredient in ingredients_data[category]:
        markup.add(telebot.types.KeyboardButton(ingredient))

    # Добавление клавиши "Назад"
    markup.add(telebot.types.KeyboardButton("Назад"))

    bot.send_message(message.chat.id, f"Добавь в свои продукты из категории
'{category}':", reply_markup=markup)

# Функция для создания клавиатуры с категориями продуктов
def create_categories_markup():
    markup = telebot.types.ReplyKeyboardMarkup(resize_keyboard=True)
    for category in ingredients_data:
        markup.add(telebot.types.KeyboardButton(category))

    # Добавление кнопок "Выбрать блюдо" и "Показать ингредиенты"
    markup.row(telebot.types.KeyboardButton("Выбрать блюдо"))
    if user_ingredients:
        markup.row(telebot.types.KeyboardButton("Показать ингредиенты"))

    return markup

# Обработчик выбора ингредиентов
@bot.message_handler(func=lambda message: message.text in
sum(ingredients_data.values(), []) or message.text == "Назад")
def handle_ingredient_selection(message):
    if message.text == "Назад":
        bot.send_message(message.chat.id, "Выбери категорию продуктов:",

```

```

reply_markup=create_categories_markup())
    return

    ingredient = message.text

    # Проверка наличия ингредиента в списке пользователя
    if ingredient in user_ingredients:
        bot.send_message(message.chat.id, f"Ингредиент '{ingredient}' уже есть в вашем списке.")
    else:
        # Добавление выбранного ингредиента в список пользователя
        user_ingredients.append(ingredient)
        bot.send_message(message.chat.id, f"Ингредиент '{ingredient}' добавлен в список.")

# Функция для поиска подходящего рецепта
def find_best_recipe(user_ingredients):
    best_recipe = None
    max_common_ingredients = 0

    for recipe in recipes_data:
        common_ingredients = len(set(user_ingredients) & set(recipe['ingredients'])) # &
множеств

        if common_ingredients > max_common_ingredients:
            max_common_ingredients = common_ingredients
            best_recipe = recipe

    return best_recipe

# Функция для поиска недостающих ингредиентов
def find_missing_ingredients(user_ingredients, recipe_ingredients):
    user_ingredients_counter = Counter(user_ingredients)
    recipe_ingredients_counter = Counter(recipe_ingredients)

    missing_ingredients = (recipe_ingredients_counter -
user_ingredients_counter).elements()
    return list(missing_ingredients)

# Обработчик команды /view_ingredients
@bot.message_handler(commands=['view_ingredients'])
def handle_view_ingredients(message):
    if not user_ingredients:
        bot.send_message(message.chat.id,
            "Ваш список продуктов пока пуст. Добавьте ингредиенты с помощью
/ingredients.")
    else:
        ingredients_str = ", ".join(user_ingredients)
        bot.send_message(message.chat.id, f"Ваши текущие ингредиенты:
{ingredients_str}")

# Обработчик команды /remove_ingredient
@bot.message_handler(commands=['remove_ingredient'])
def handle_remove_ingredient(message):
    if not user_ingredients:
        bot.send_message(message.chat.id,
            "Ваш список продуктов пока пуст. Добавьте ингредиенты с помощью
/ingredients.")
    else:
        removed_ingredient = user_ingredients.pop()
        bot.send_message(message.chat.id, f"Ингредиент '{removed_ingredient}' удален из
вашего списка.")

```

```

# Обработчик команды "Показать ингредиенты"
@bot.message_handler(func=lambda message: message.text == "Показать ингредиенты" and
user_ingredients)
def handle_show_ingredients(message):
    handle_view_ingredients(message)

# Обработчик команды "Выбрать блюдо"
@bot.message_handler(func=lambda message: message.text == "Выбрать блюдо")
def handle_cook(message):
    bot.send_message(message.chat.id, "Ищем подходящий рецепт...")

    # Находим подходящий рецепт
    best_recipe = find_best_recipe(user_ingredients)

    if best_recipe:
        bot.send_message(message.chat.id, f"Предлагаю приготовить блюдо:
{best_recipe['name']}")
        missing_ingredients = find_missing_ingredients(user_ingredients,
best_recipe['ingredients'])

        if missing_ingredients:
            bot.send_message(message.chat.id, f"Не хватает следующих ингредиентов: {'',
'.join(missing_ingredients)}")
        else:
            bot.send_message(message.chat.id, f"Ингредиентов достаточно, можем
приступить!")
    else:
        bot.send_message(message.chat.id, "Извините, не удалось найти подходящий
рецепт.")

# Обработчик команды /clear
@bot.message_handler(commands=['clear'])
def handle_clear(message):
    # Очистка списка user_ingredients
    user_ingredients.clear()
    bot.send_message(message.chat.id, "Список ингредиентов очищен.")

# Обработчик команды /cook
@bot.message_handler(commands=['cook'])
def handle_cook(message):
    bot.send_message(message.chat.id, "Ищем подходящий рецепт...")

    # Находим подходящий рецепт
    best_recipe = find_best_recipe(user_ingredients)

    if best_recipe:
        bot.send_message(message.chat.id, f"Предлагаю приготовить блюдо:
{best_recipe['name']}")
        missing_ingredients = find_missing_ingredients(user_ingredients,
best_recipe['ingredients'])

        if missing_ingredients:
            bot.send_message(message.chat.id, f"Не хватает следующих ингредиентов: {'',
'.join(missing_ingredients)}")
        else:
            bot.send_message(message.chat.id, "Извините, не удалось найти подходящий
рецепт.")

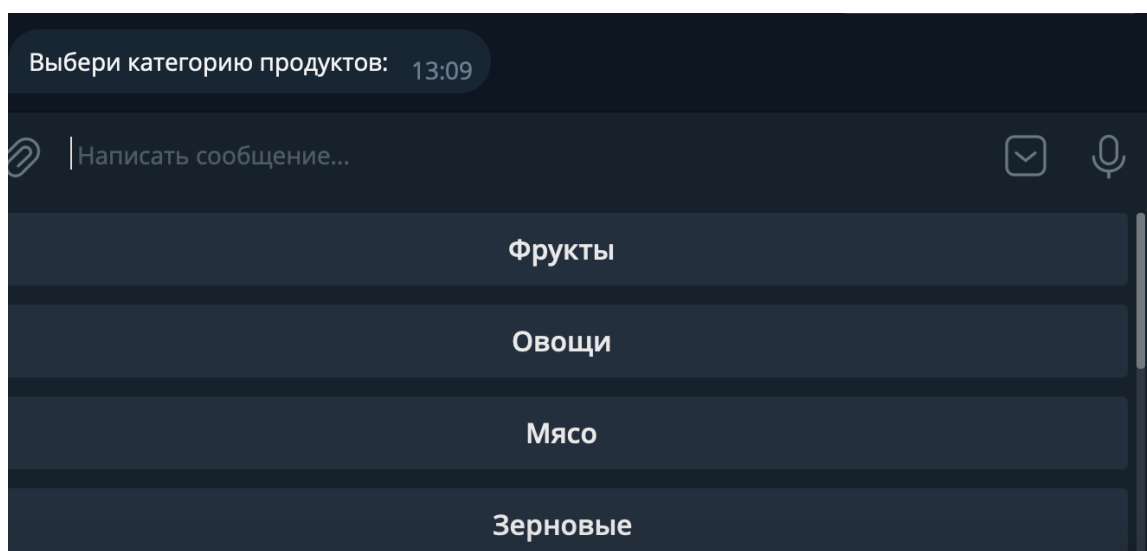
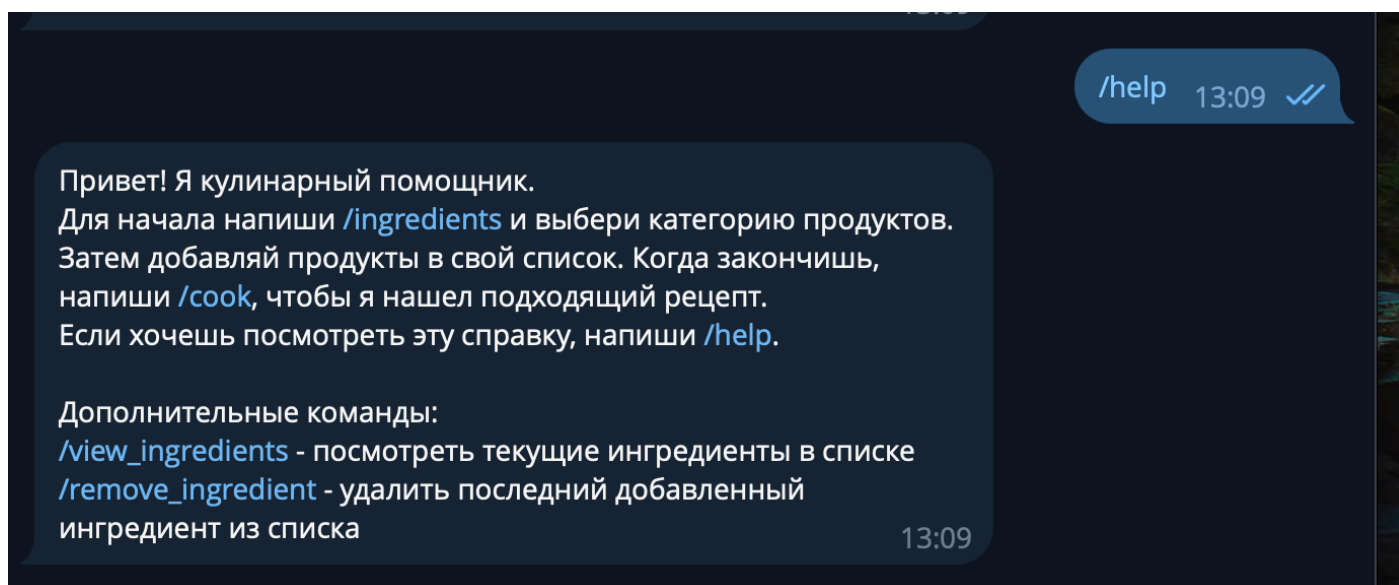
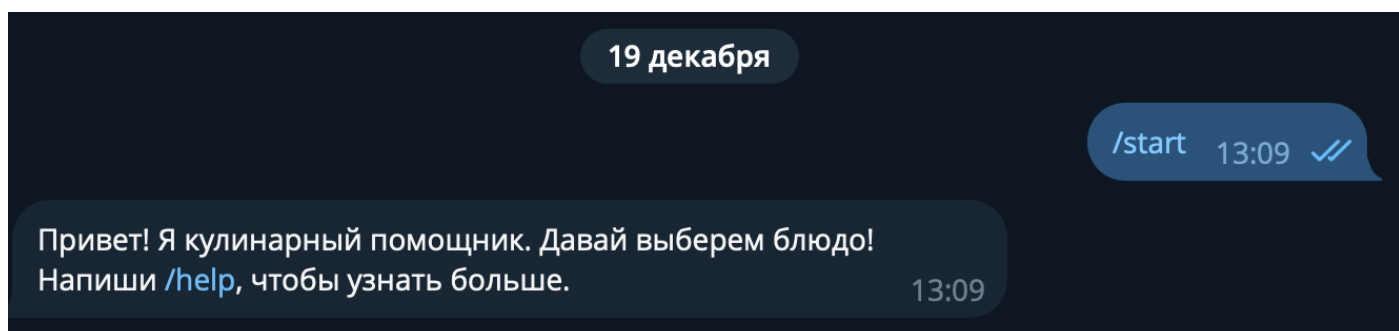
# Обработчик неизвестных команд
@bot.message_handler(func=lambda message: True)
def handle_unknown(message):
    bot.send_message(message.chat.id, "Извините, я не понимаю ваш запрос. Для справки

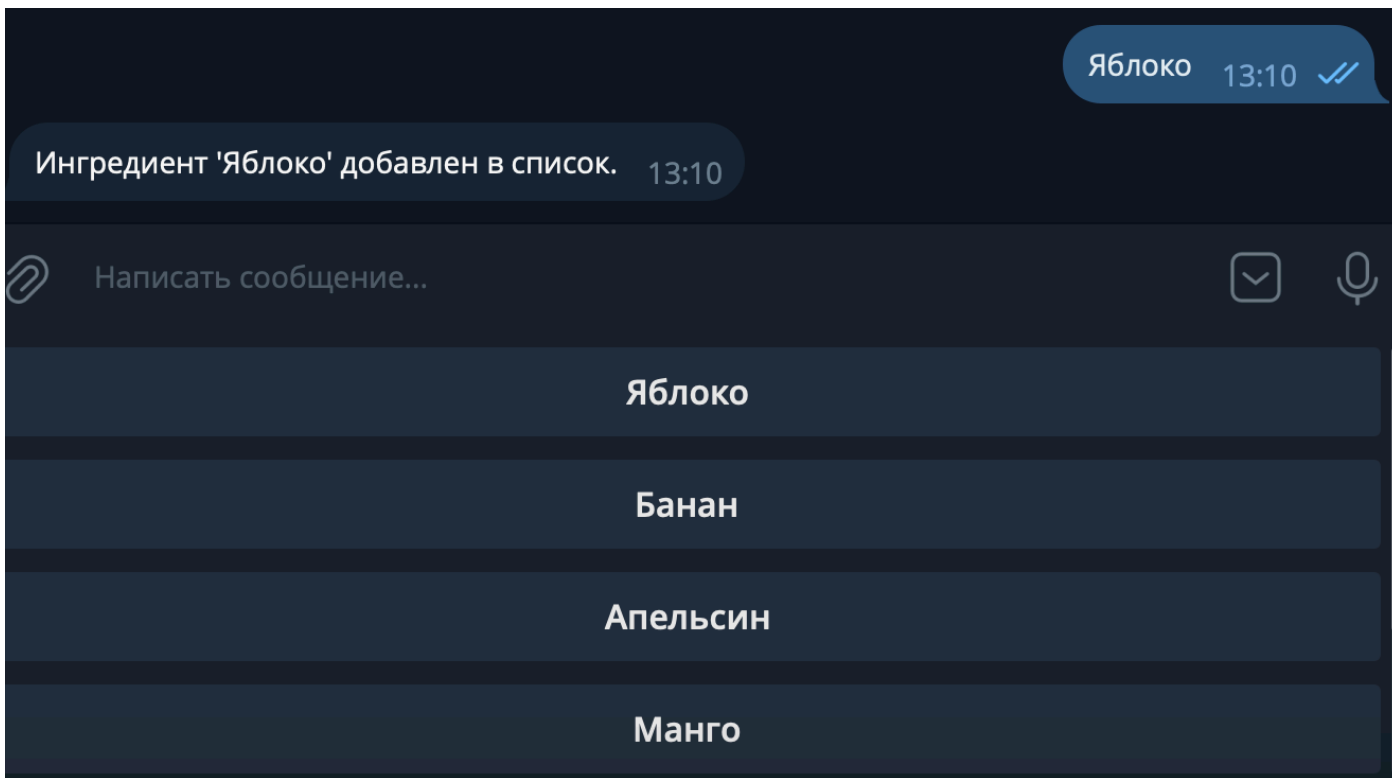
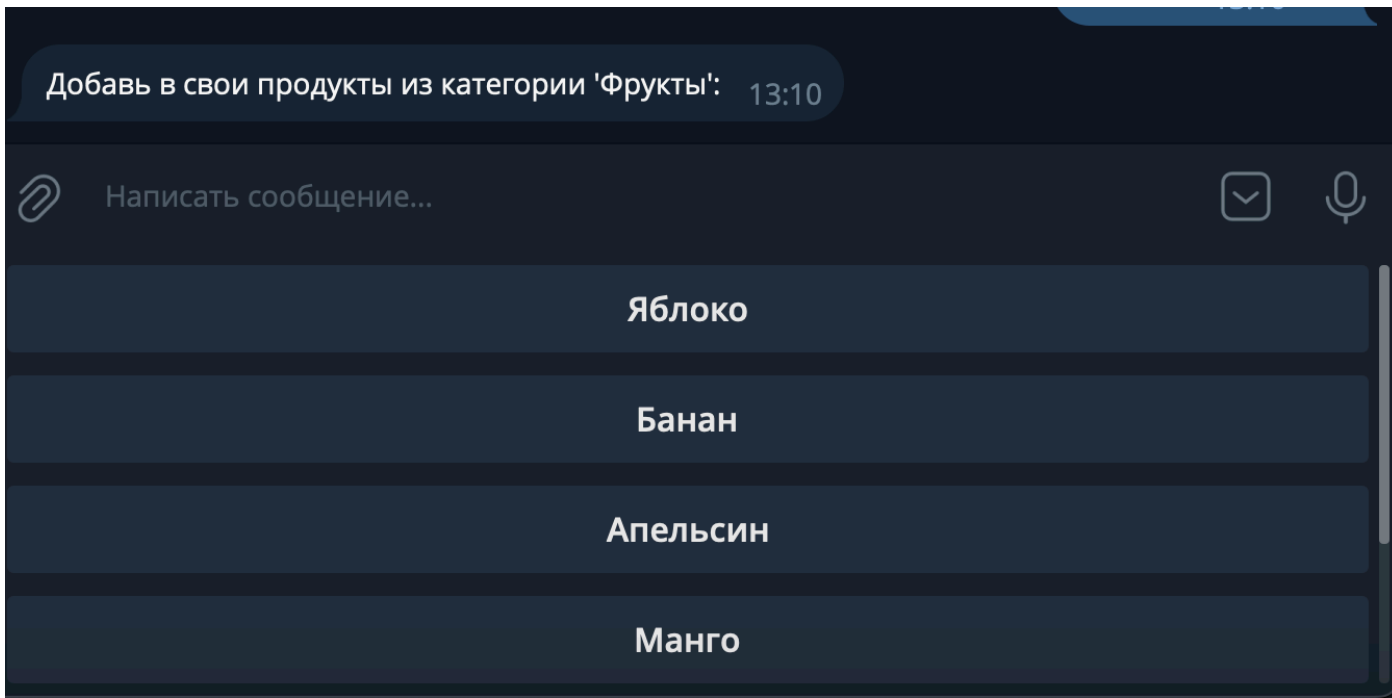
```

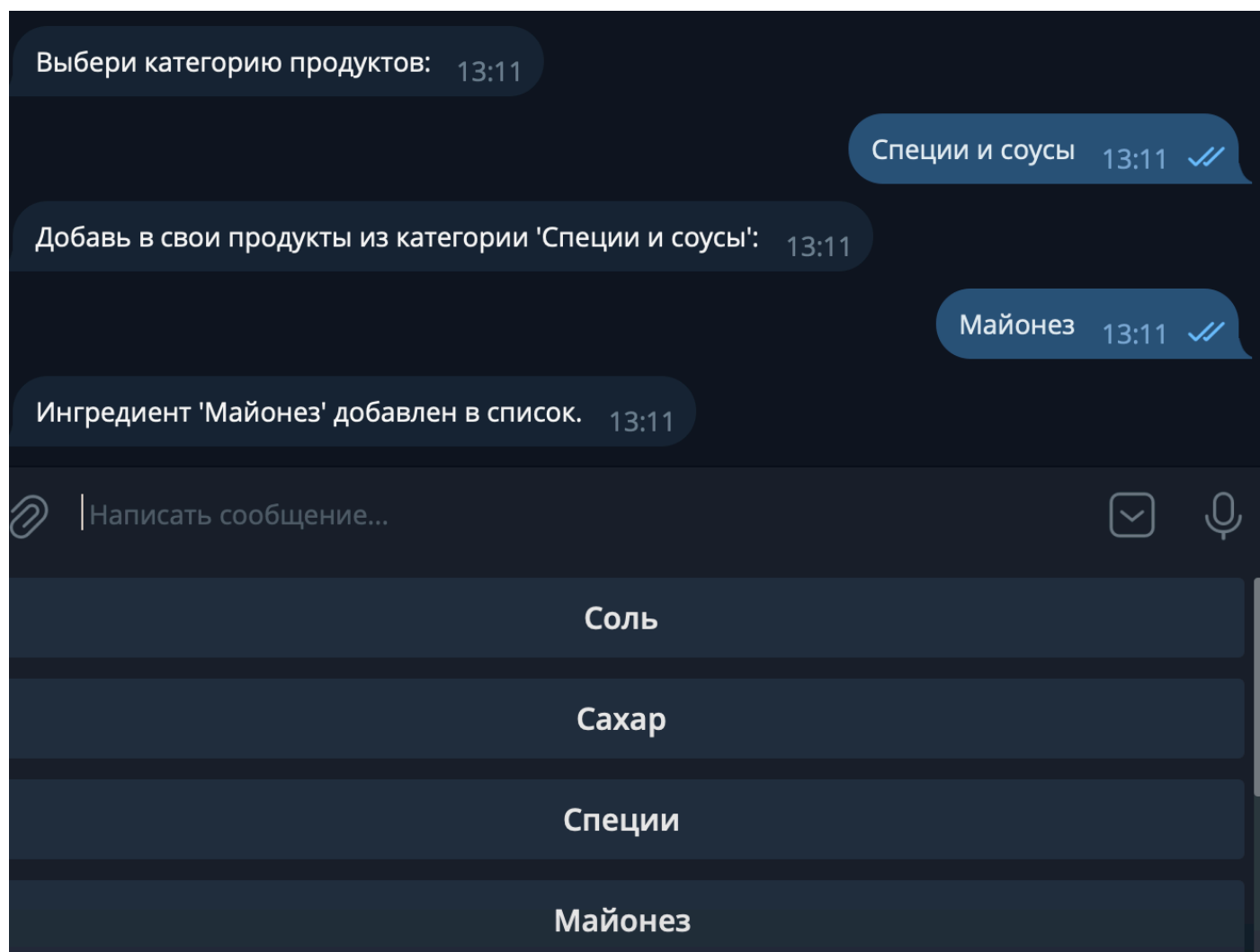
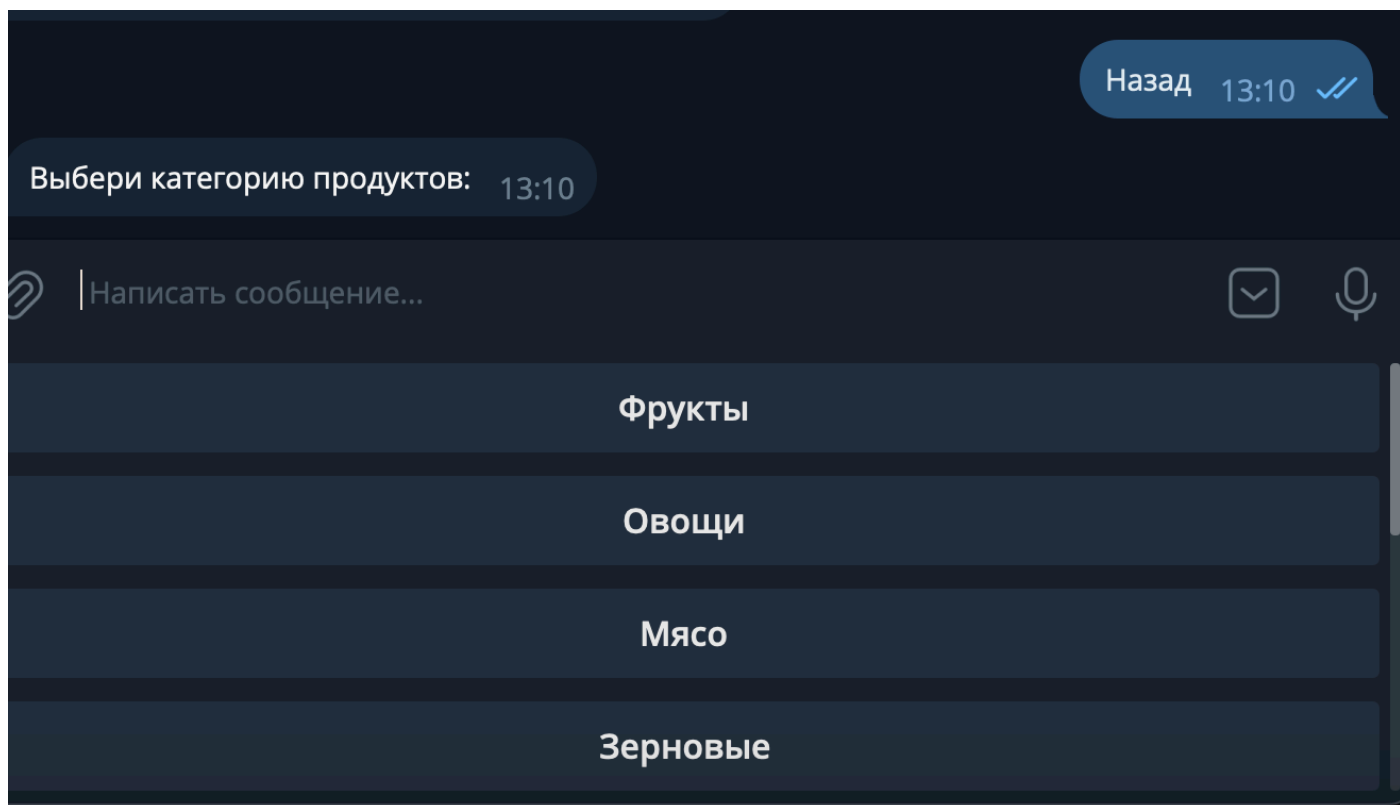
```
напишите /help.")

if __name__ == "__main__":
    bot.polling(none_stop=True)
```

Пример работы







Выбери категорию продуктов: 13:10

Мясо 13:10 ✓✓

Добавь в свои продукты из категории 'Мясо': 13:10

Курица 13:10 ✓✓

Ингредиент 'Курица' добавлен в список. 13:10



Написать сообщение...



Курица

Говядина

Свинина

Креветки

Выбери категорию продуктов: 13:11

Показать ингредиенты 13:11 ✓✓

Ваши текущие ингредиенты: Яблоко, Курица, Майонез 13:11



Написать сообщение...

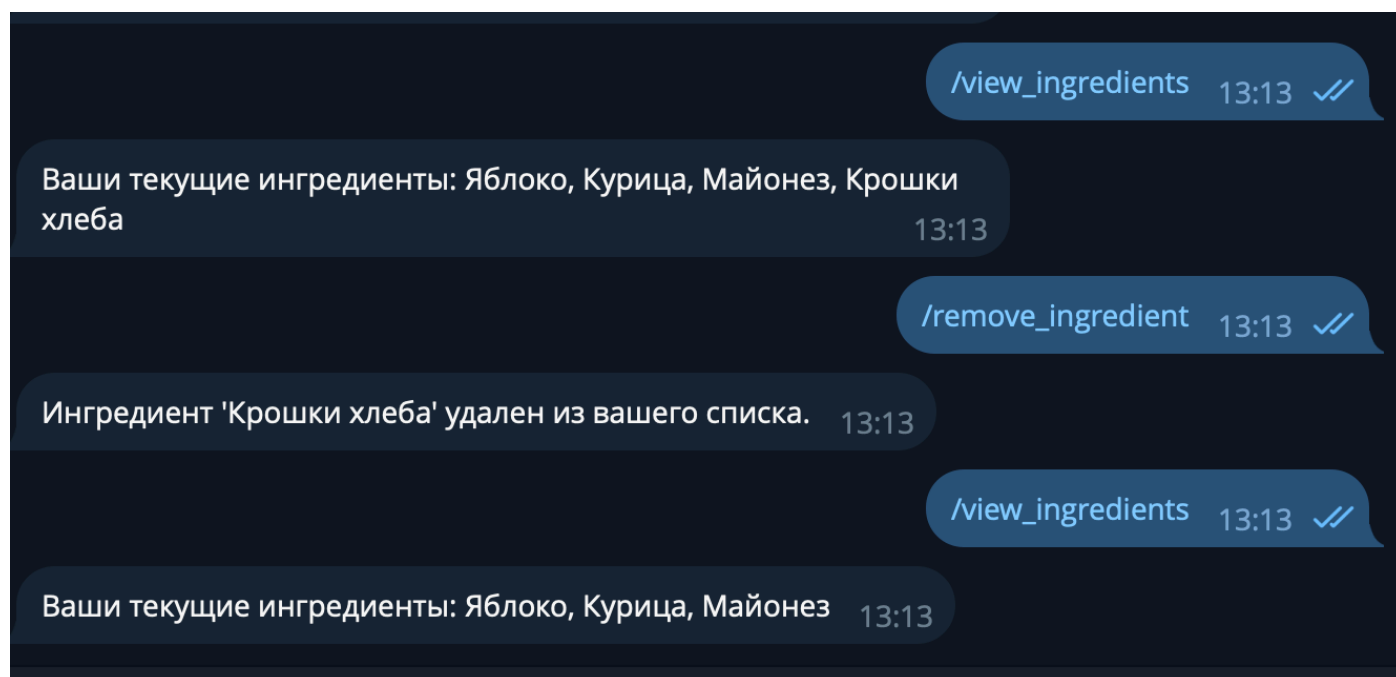
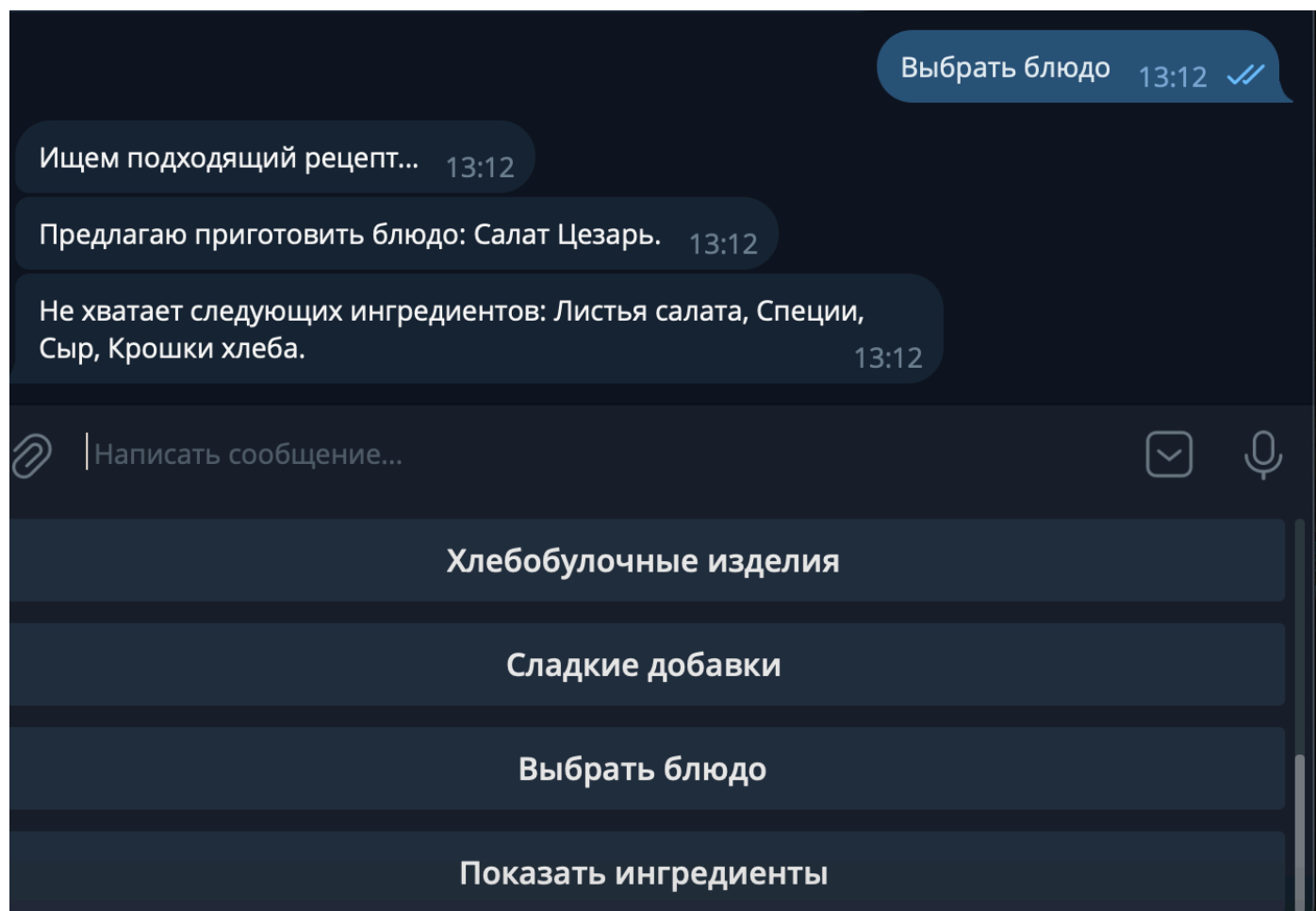


Хлебобулочные изделия

Сладкие добавки

Выбрать блюдо

Показать ингредиенты



Яблоко 13:13 ✓✓

Ингредиент 'Яблоко' уже есть в вашем списке. 13:13



| Написать сообщение...



Яблоко

Банан

Апельсин

Манго