Лабораторная работа 6. Анпилов Кирилл ПИ21-4.

Цель работы

Провести написать автотесты для телеграм-бота для учета продуктов с использованием предварительно разработанной тестовой документации для проверки функциональности системы, выявления дефектов и обеспечения соответствия требованиям.

Описание реализованных автотестов

Так как у нас телеграм-бот у нас есть два пути для тестирования приложения. Первый, как указано в лабораторной работе мы можем тестировать бота, как веб-приложение, зайдя в веб-версию телеграм-клента, через selenium.

Второй же – лучший подход использовать api телеграмма и специальные библиотеки для общения с api.

**Автотесты:**

|  |  |  |
| --- | --- | --- |
| № | Input\_values | Output\_values |
| 1 | /start | Привет! Я – Kara и меню |
| 2 | /help | Пишите @anpilove |

Код автотестов

import pytest  
from unittest.mock import AsyncMock  
from aiogram import types, Bot, Dispatcher  
from handlers.user\_handlers import process\_start\_command, process\_help\_command  
from lexicon.lexicon\_ru import LEXICON\_RU  
from keyboards.keyboards import menu\_kb  
  
  
async def test\_main(dp):  
 await test\_start\_command(dp)  
 await test\_help\_command(dp)  
 print('Тесты успешно пройдены')  
  
  
@pytest.mark.asyncio  
async def test\_start\_command(dp):  
 mock\_message = AsyncMock()  
 # mock\_message = MockMessage("/start", chat\_id=12345)  
 # await dp.message\_handlers.notify(mock\_message)  
  
 await process\_start\_command(mock\_message)  
 mock\_message.answer.assert\_called\_with(text=LEXICON\_RU['/start'],  
 reply\_markup=menu\_kb)  
  
  
@pytest.mark.asyncio  
async def test\_help\_command(dp):  
 mock\_message = AsyncMock()  
 await process\_help\_command(mock\_message)  
 mock\_message.answer.assert\_called\_with(text=LEXICON\_RU['/help'])

Отчет о тестировании

|  |  |  |  |
| --- | --- | --- | --- |
| № | Input\_values | Output\_values | Стутус |
| 1 | /start | Привет! Я – Kara и меню | Успешно |
| 2 | /help | Пишите @anpilove | Успешно |

При ошибке в тестировании пишет

A screen shot of a computer code

Description automatically generated

Если все хорошо и наш бот запускается

A screen shot of a computer

Description automatically generated

Выводы по работе

Для себя я открыл автоматическое тестирование, в ходе которого действительно нашел ошибку в своем простом коде. Там, где не ожидал.

Считаю это необходимым модулем в среднем-большем проекте, который сохранит время и нервы и позволит быть уверенным в запущенном продукте.

Список используемых источников

1. <https://medium.com/@duketemon/how-to-test-a-telegram-bot-ba54eb1cadc0>
2. <https://stackoverflow.com/questions/36891407/use-telegram-client-for-bot-testing-not-bot-api>