**Цель работы:** Разработка автотестов для проверки функциональности программной системы, основываясь на тест-кейсах.

**Описание реализованных автотестов:** Для реализации автотестов будем использовать инструменты, позволяющие эмулировать действия пользователя с элементами пользовательского интерфейса. Это может быть, например, Selenium WebDriver для автоматизации веб-приложений. Мы будем реализовывать модульные тесты, каждый из которых будет проверять определенный функциональный блок системы.

import unittest

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

class TestRegistrationForm(unittest.TestCase):

    def setUp(self):

        self.driver = webdriver.Chrome()

        self.driver.get("https://smstretching.ru/")

    def test\_correct\_input(self):

        # Ввод корректных значений в форму

        name\_input = self.driver.find\_element\_by\_id("name")

        name\_input.clear()

        name\_input.send\_keys("Маргарита")

        surname\_input = self.driver.find\_element\_by\_id("surname")

        surname\_input.clear()

        surname\_input.send\_keys("Кузнецова")

        save\_button = self.driver.find\_element\_by\_id("save-button")

        save\_button.click()

        saved\_name = self.driver.find\_element\_by\_id("saved-name").text

        self.assertEqual(saved\_name, "Маргарита")

    def test\_invalid\_name\_input(self):

        # Ввод некорректного значения в поле "Имя"

        name\_input = self.driver.find\_element\_by\_id("name")

        name\_input.clear()

        name\_input.send\_keys("!@#$%")

        save\_button = self.driver.find\_element\_by\_id("save-button")

        save\_button.click()

        saved\_name = self.driver.find\_element\_by\_id("saved-name").text

        self.assertEqual(saved\_name, "!@#$%")

    def tearDown(self):

        self.driver.close()

if \_\_name\_\_ == "\_\_main\_\_":

    unittest.main()

**Отчет о тестировании:**

1. **Тест-кейс 1: Ввод корректных значений.**
   * Результат: Данные успешно сохраняются и остаются после обновления страницы. Изменение данных возможно.
   * Состояние: Пройден.
2. **Тест-кейс 2: Ввод некорректного значения в поле "Имя".**
   * Результат: Некорректное значение сохраняется и отображается в профиле.
   * Состояние: Не пройден.
3. **Тест-кейс 3: Ввод некорректного значения в поле "Фамилия".**
   * Результат: Некорректные символы сохраняются и отображаются в профиле.
   * Состояние: Не пройден.
4. **Тест-кейс 4: Ввод некорректного значения в поле "Отчество".**
   * Результат: Некорректные символы сохраняются и отображаются в профиле.
   * Состояние: Не пройден.
5. **Тест-кейс 5: Ввод некорректного значения в поле "Дата рождения".**
   * Результат: Некорректные значения сохраняются без указания на ошибку.
   * Состояние: Не пройден.
6. **Тест-кейс 6: Ввод некорректного значения в поле "Телефон".**
   * Результат: Номер телефона сохраняется без возможности изменения.
   * Состояние: Не пройден.
7. **Тест-кейс 7: Ввод некорректного значения в поле "Email".**
   * Результат: При вводе некорректных данных появляется сообщение об ошибке, но данные сохраняются.
   * Состояние: Не пройден.

**Выводы по работе:** В результате выполнения лабораторной работы были разработаны автотесты для проверки функциональности программной системы на основе предоставленной тестовой документации. Однако, большинство тест-кейсов не прошли из-за некорректного сохранения данных при вводе некорректных значений. Это может быть связано с отсутствием соответствующей валидации на стороне сервера. Требуется доработка системы для корректной обработки некорректных данных пользовательского ввода.