

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА СИСТЕМЫ СРАВНЕНИЯ СКРИНШОТОВ ДЛЯ
АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ВИЗУАЛЬНОЙ
ЧАСТИ ВЕБ-ПРИЛОЖЕНИЙ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Анохина Данилы Дмитриевича

Научный руководитель

зав. каф., к. ф.-м. н., доцент

С. В. Миронов

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2025

ВВЕДЕНИЕ

В последние годы разработка веб-приложений стала все более сложной и требовательной к качеству. Одним из ключевых аспектов обеспечения качества веб-приложений является автоматизированное тестирование, которое позволяет быстро и эффективно проверять функциональность и визуальную часть приложения. Скриншотное тестирование является одним из видов автоматизированного тестирования, который позволяет сравнивать визуальное представление веб-приложения с эталонными скриншотами.

Актуальность темы разработки системы сравнения скриншотов для автоматизированного тестирования визуальной части веб-приложений обусловлена необходимостью создания эффективных и автоматизированных методов тестирования, которые позволяют сократить время и ресурсы, затрачиваемые на тестирование. Кроме того, скриншотное тестирование может быть использовано для проверки веб-приложений на различных устройствах и платформах, что является важным аспектом обеспечения кросс-браузерной и кросс-платформенной совместимости.

Целью данной работы является разработка системы сравнения скриншотов для автоматизированного тестирования визуальной части веб-приложений, которая позволит автоматизировать процесс тестирования и сократить время и ресурсы, затрачиваемые на тестирование.

Для достижения этой цели необходимо решить следующие задачи:

- Изучить теоретические основы тестирования программного обеспечения;
- Исследовать инструменты и фреймворки для автоматизации тестирования на Python;
- Проанализировать функциональность объекта тестирования;
- Выбрать технологический стек;
- Разработать архитектуру системы;
- Реализовать демонстрационный тест-кейс;
- Оценить эффективность предложенных решений.

В работе будут рассмотрены основные теоретические аспекты тестирования программного обеспечения, а также практические вопросы автоматизации скриншотного тестирования веб-приложений. Особое внимание будет уделено процессу сравнения эталонных скриншотов и скриншотов, сделанных в результате выполнения тест-кейсов.

Бакалаврская работа состоит из обозначений и сокращений, определений, введения, 2 разделов, заключения, списка использованных источников и 19 приложений. Общий объем работы — 90 страниц, из них 44 страницы — основное содержание, включая 10 рисунков, список использованных источников информации — 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Раздел 1 данной работы посвящен основным сведениям теории тестирования, необходимым для лучшего понимания специфики работы. В частности, в подразделе 1.1 приведены определения, цели, виды и принципы тестирования. Также рассмотрена роль тестировщика и автоматизация тестирования. Все это необходимо для рассмотрения темы работы.

В подразделе 1.2 рассматриваются основные концепции тестирования и артефакты, такие как тест-кейсы, тестовые сценарии, тестовые данные и метрики.

Подраздел 1.3 отвечает на вопрос, зачем нужна автоматизация тестирования, и какие существуют техники автоматизации. Также рассматривается самый популярный фреймворк для автоматизации тестирования на Python — PyTest.

В подразделе 1.4 исследуются техники скриншотного тестирования, инструменты и библиотеки для него, которые будут использованы в дальнейшем в настоящей работе.

В разделе 2 представлена практическая часть работы. В частности, в подразделе 2.1 описан объект тестирования.

Объектом тестирования является ключевой продукт компании «Positive Technologies» — «MaxPatrol SIEM», который, в свою очередь, представляет собой систему выявления инцидентов ИБ и попыток нарушения киберустойчивости компании. В частности, будем рассматривать страницу логина, показанную на рисунке 1.

pt Management and Configuration

Вход на площадку Площадка

☐ Запомнить меня

Светлая



Рисунок 1 – Страница логина веб-приложения «MaxPatrol SIEM»

В подразделе 2.2.1 приведен технологический стек, который использован в проекте:

1. Python: версия 3.12 — основной язык программирования проекта;
2. Pytest: версия 8.3.4 — тестовый фреймворк;
3. Playwright: версия 0.7.0 — это фреймворк для автоматизированного тестирования веб-приложений, который позволяет симулировать взаимодействие пользователя с веб-страницей;
4. Allure: версия 2.13. — библиотека для создания отчетов о тестировании;
5. NumPy: версия 2.2.3 — библиотека для работы с числовыми данными;
6. OpenCV: версия 4.11.0.86 — библиотека для работы с изображениями и компьютерным зрением;
7. Scikit-image: версия 0.25.2 — библиотека для работы с изображениями и компьютерным зрением.

В подразделе 2.2.2 представлена общая схема архитектуры проекта, которая иллюстрирует взаимосвязи между различными компонентами и модулями системы сравнения скриншотов, как показано на рисунке 2.

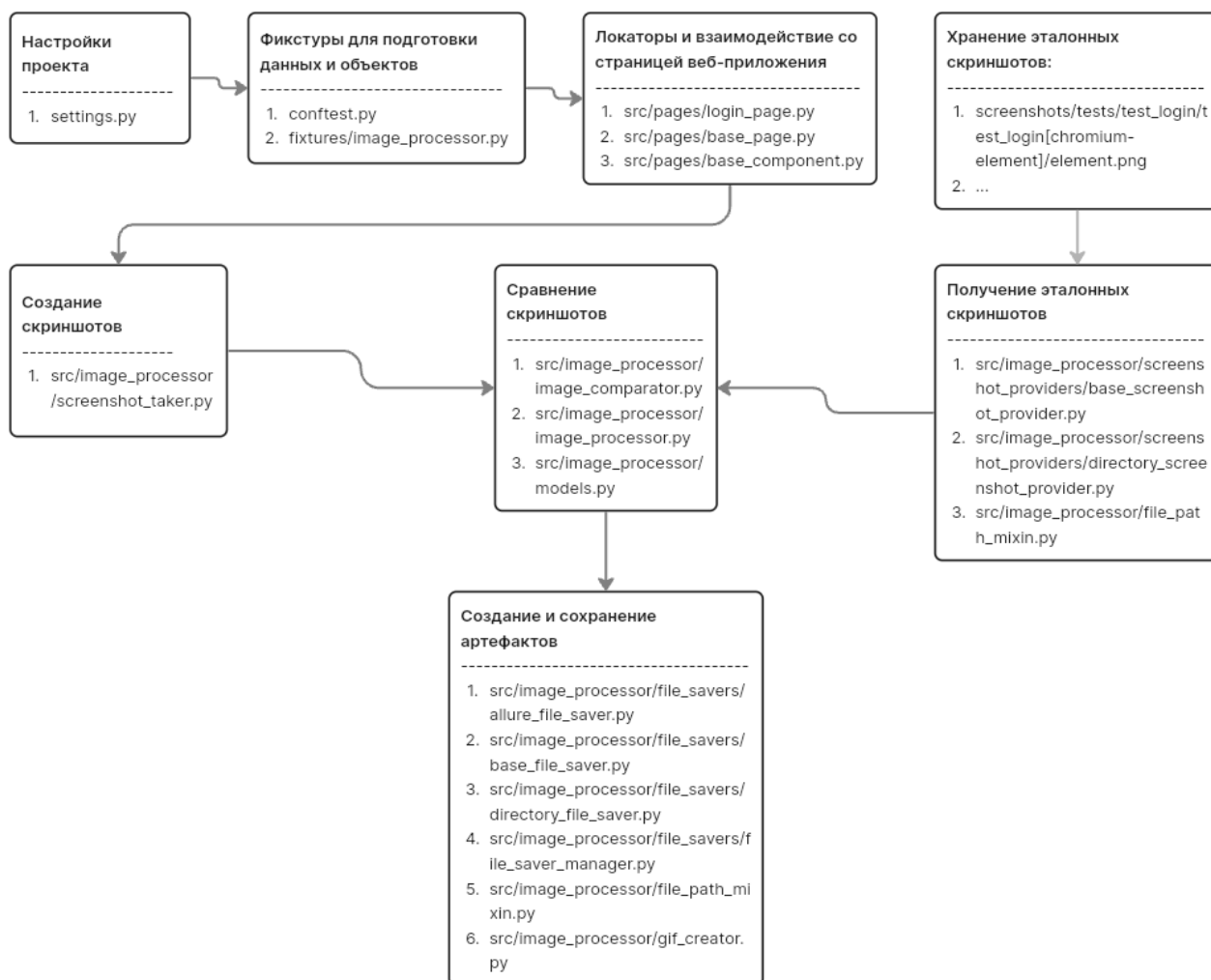


Рисунок 2 – Общая схема архитектуры проекта

Далее в подразделах 2.2.3-2.2.10 подробно описаны различные логические модули проекта, включая настройки проекта (2.2.3), фикстуры для подготовки данных и объектов (2.2.4), хранение эталонных скриншотов (2.2.5), локаторы и взаимодействие со страницей веб-приложения (2.2.6), создание скриншотов (2.2.7), получение эталонных скриншотов (2.2.8), сравнение скриншотов (2.2.9) и создание и сохранение артефактов (2.2.10).

Подробнее можно остановиться на алгоритме сравнения скриншотов:

1. Преобразование скриншотов из байтов в формат OpenCV;
2. Преобразование скриншотов в градации серого;
3. Проверка форм скриншотов;
4. Вычисление SSIM;
5. Обнаружение различий;
6. Получение контуров;
7. Выделение контуров;

8. Преобразование результатов в байты;
9. Сохранение артефактов.

В подразделе 2.3 описана реализация демонстрационного тест-кейса. Тест-кейс реализован для демонстрации возможностей проекта по сравнению скриншотов и проверяет корректность системы на примере страницы авторизации.

Для демонстрации работы были сделаны эталонные скриншоты с помощью модуля `src/image_processor/screenshot_taker.py`:

1. success — вся страница авторизации показана на рисунке 3;



Рисунок 3 – Эталонный скриншот страницы авторизации

2. fail — вся страница авторизации показана на рисунке 3;
3. mask — вся страница авторизации с маской на поля ввода логина и пароля показана на рисунке 4;

Вход на площадку Площадка

☐ Запомнить меня[Как в системе](#)

Рисунок 4 – Эталонный скриншот страницы авторизации с маской

4. element — форма логина показана на рисунке 5.

Вход на площадку Площадка

☐ Запомнить меня

Рисунок 5 – Эталонный скриншот формы логина

Тест-кейс также параметризован на 4 случая в зависимости от эталонного скриншота.

Шаги выполнения тест-кейса:

1. Открытие страницы авторизации;
2. Ожидание видимости полей ввода логина и пароля, а также чекбокса "Запомнить меня";
3. Выбор элемента для скриншота, который может быть либо страницей, либо страницей с маской, либо отдельным элементом (в зависимости от

параметризации);

4. Если параметризация равна fail, делаем дополнительные шаги, чтобы сравнение с эталонным скриншотом закончилось отрицательно: снимаем фокус с поля ввода логина и ставим галочку в чекбокс "Запомнить меня";
5. Делаем скриншот и передаем данные в image_processor;
6. Если результат отрицательный, вызываем ошибку "Ожидаемый и эталонный скриншоты не совпадают".

В подразделе 2.4 запускаются и анализируются результаты выполнения тест-кейса. Тесты прошли успешно за исключением одного test_login[chromium-fail], который и должен был упасть, потому что эталонный скриншот не соответствует текущему. Результат выполнения тестов представлен на рисунке 6.



Рисунок 6 – Результаты выполнения тестов

Как видно из рисунка, тест test_login[chromium-fail] упал, что означает, что скриншот не соответствует эталонному. В корне проекта создавалась директория с артефактами, которая содержит результаты тестирования. Директория имеет следующую структуру: test-results -> tests -> test_login(название теста) -> test_login[chromium-fail](параметризация теста).

В этой директории находятся три файла:

1. fail - анимированный gif с рамками отличий.gif
2. fail - базовый скриншот с рамками отличий.png
3. fail - новый скриншот с рамками отличий.png

Рисунок 7 представляет собой базовый скриншот с рамками отличий, а рисунок 8 представляет собой новый скриншот с рамками отличий. Эти скриншоты показывают различия между эталонным и текущим скриншотами. Также генерируется GIF файл.

Вход на площадку | Площадка

Логин

Пароль

Запомнить меня

Войти

Как в системе



Рисунок 7 – Базовый скриншот с рамками отличий

Вход на площадку | Площадка

Логин

Пароль

☒ Запомнить меня

Войти

Как в системе



Рисунок 8 – Новый скриншот с рамками отличий

Такие артефакты создаются для каждого теста, что позволяет проанализировать результаты тестирования и понять, почему тест упал.

Кроме того, также был сгенерирован allure отчет, который содержит информацию о результатах тестирования. Рисунок 9 представляет собой скрин-

шот allure отчета, а рисунок 10 представляет собой скриншот того, что к нему приложен GIF файл.

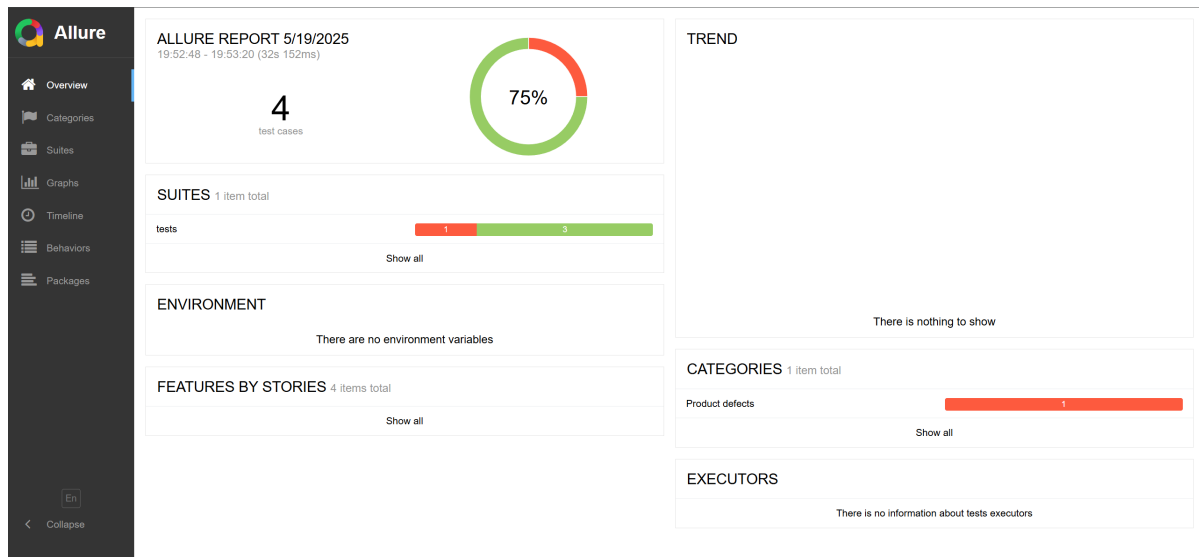


Рисунок 9 – Allure отчет

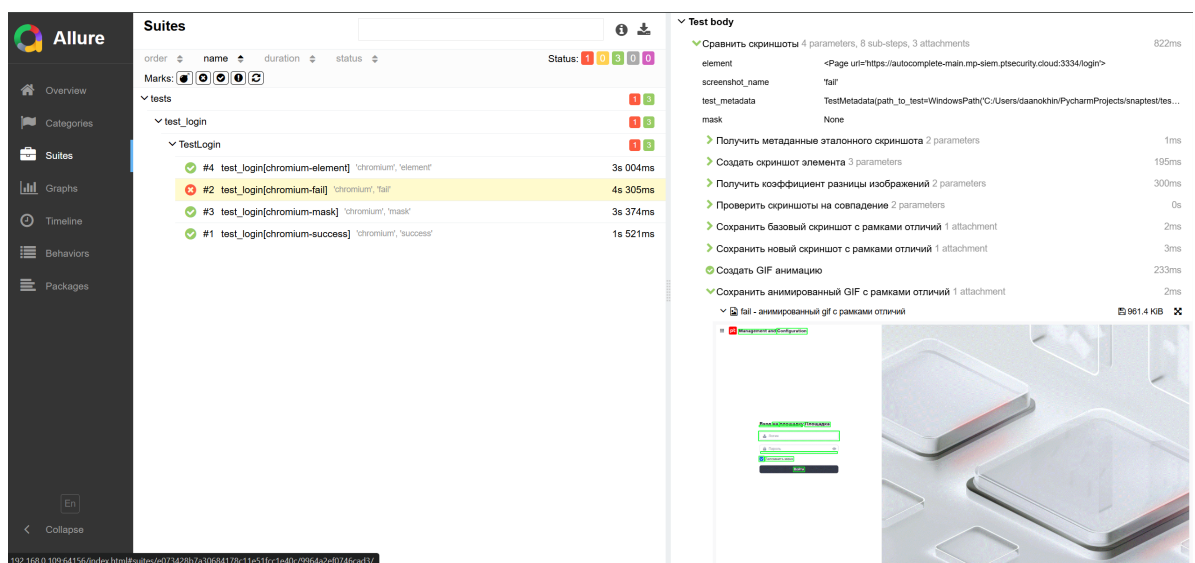


Рисунок 10 – Приложенный GIF файл в тесте

ЗАКЛЮЧЕНИЕ

В данной работе была разработана система сравнения скриншотов для автоматизированного тестирования визуальной части веб-приложений. Основной целью работы было создание эффективных и автоматизированных методов тестирования, которые позволяют сократить время и ресурсы, затрачиваемые на тестирование.

В результате работы были достигнуты следующие результаты:

- Была разработана архитектура системы сравнения скриншотов, которая включает в себя модули для сравнения скриншотов, сохранения скриншотов и управления процессом тестирования;
- Были реализованы фикстуры для подготовки данных и объектов, которые позволяют упростить процесс тестирования и сделать его более эффективным;
- Были реализованы алгоритмы сравнения скриншотов, которые позволяют обнаруживать различия между эталонными скриншотами и скриншотами, сделанными в результате выполнения тест-кейсов;
- Была разработана система для сохранения скриншотов, которая позволяет сохранять скриншоты в различных форматах и директориях.

Оценка полноты решений поставленных задач показала, что все задачи были решены в полном объеме. Были изучены теоретические основы тестирования программного обеспечения, исследованы инструменты и фреймворки для автоматизации тестирования на Python, проанализирована функциональность объекта тестирования, выбран технологический стек, разработана архитектура системы, реализован демонстрационный тест-кейс и оценена эффективность предложенных решений.

Рекомендации по использованию результатов работы включают в себя использование системы сравнения скриншотов для автоматизированного тестирования визуальной части веб-приложений.

Оценка эффективности предложенных решений показала, что система сравнения скриншотов является эффективным инструментом для автоматизированного тестирования визуальной части веб-приложений. Система позволяет сократить время и ресурсы, затрачиваемые на тестирование, и обеспечивает высокую точность обнаружения различий между эталонными скриншотами и скриншотами, сделанными в результате выполнения тест-кейсов.