Лабораторная работа 8

Тема лабораторной работы: работа с классификацией видов тестирования.

Цель работы: научиться классифицировать виды тестирования на примере выбранного программного продукта.

## Программный продукт

Сайт для стилизации изображений ДЕН Ь.

## Классификация тестирования

Виды тестирования

1. Функциональное тестирование

Проверка функционала приложения: загрузка изображения, получение готового результата, бегунки и различные кнопки.

1. Нефункциональное тестирование

Производительное тестирование: оценка скорости работы сайта по всему функционалу.

Тестирование удобства использования: проведение тестов с целью выявления слабых мест в дизайне и структуре страниц сайта.

Тестирование совместимости: проврека работы приложения в разных условиях: планшет, телефон, другой браузер.

1. Регрессионное тестирование

Проверка, что новые изменения в коде не нарушили существующую функциональность.

1. Приемочное тестирование

Проверка, что приложение соответствует ожиданиям и требованиям конечных пользователей.

**Типы тестирования**

1. Черный ящик (Black Box Testing)

◦ Функциональное тестирование пользовательских сценариев без знания внутренней структуры кода.

2. Белый ящик (White Box Testing)

◦ Проверка внутренней логики и структуры кода, может быть применено для модульных тестов.

3. Серый ящик (Gray Box Testing)

◦ Тестирование с ограниченными знаниями о внутренней структуре, может быть использовано для интеграционных тестов.

**Методы тестирования**

Методика ручного тестирования включает выполнение тестов в соответствии с написанными заранее сценариями, а также проверку интерфейса и пользовательских сценариев.

1. Для автоматизации функциональных тестов часто применяется Selenium WebDriver.
2. Интеграционное тестирование направлено на проверку взаимодействия компонентов приложения, как, например, фронтенд и бэкенд.
3. Тестирование на основе сценариев предполагает выполнение тестов с использованием заранее определенных сценариев использования системы.

**Уровни тестирования**

1. Модульное тестирование (Unit Testing)

◦ Тестирование отдельных функций или методов, особенно если они содержат сложную логику обработки изображений.

2. Интеграционное тестирование (Integration Testing)

◦ Проверка взаимодействия между различными модулями приложения, такими как загрузка изображений, применение фильтров и сохранение результатов.

3. Системное тестирование (System Testing)

◦ Полное тестирование всей системы целиком, включая все интегрированные модули и компоненты.

4. Приемочное тестирование (Acceptance Testing)

◦ Финальное тестирование для проверки соответствия системы требованиям и ожиданиям пользователей.

В данном случае основное внимание будет уделено функциональному тестированию, производительному тестированию и тестированию удобства использования. Типы тестирования включают использование черного ящика с помощью Selenium и ручные тесты для проверки сценариев, а методы включают автоматизированное тестирование с использованием Selenium WebDriver и ручное тестирование для сценариев, требующих субъективной оценки.

## Примеры дефектов

Тестирование функциональности

1. Неправильная работа функций
2. Проблемы с фильтрами и стилизацией изображений, не соответствующие спецификации.
3. Ошибки при применении фильтров к изображениям.
4. Сбои при загрузке и сохранении изображений из-за неподдерживаемых форматов или неправильного сохранения.

Тестирование производительности

1. Долгие временные задержки при обработке изображений с использованием фильтров.
2. Низкая производительность при большом количестве пользователей.

Тестирование удобства использования

1. Сложности с пользовательским интерфейсом, затрудняющие доступ к функциям.
2. Проблемы с отображением интерфейса на различных устройствах.

Testing compatibility

1. Плохое отображение или неработающие функции в определенных браузерах.
2. Ошибки на мобильных устройствах.

Регрессионное тестирование

1. Изменения в коде ломают существующие фильтры.
2. Добавление новых функций приводит к сбоям в уже работающих сценариях.

Приемочное тестирование

1. Несоответствие требованиям пользователя, например, отсутствие поддержки определенных форматов изображений.
2. Интерфейс не соответствует ожиданиям пользователей по функциональности и удобству использования.

## Выводы

Применение различных видов тестов позволяет провести комплексную оценку качества, а также выявить слабые места в приложении.