# **5** ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

Тестирование программного обеспечения – процесс исследования программного продукта на конечном наборе тестов, направленных на выявления ошибок и показывающий соответствие реальный продукт с ожидаемым. Так же глобально тестирование программного обеспечения предоставляется информацию о качестве программного продукта, включает в себя действия по планированию работ и проектированию тестов, анализ полученных результатов от тестов.

Тестирование бывает следующих видов:

1. Функциональное тестирование – тестирование программного обеспечения с целью проверки реализуемости функциональных требований. То есть способностью программного продукта решать поставленные ему задачи нужные пользователям.
2. Стресс тестирование – тестирование программного обеспечения, которое используется для проверки устойчивости системы к нагрузкам и определения производительности системы и предсказания поведения системы при стрессовых нагрузках.
3. Регрессионное тестирование – тестирование программного обеспечения, направленное на обнаружение ошибок в уже протестированном продукте при добавлении нового функционала.
4. Нагрузочное тестирование – один из видов тестирования производительности программного обеспечения, направленны на получение показателей временного отклика и производительности системы. Используется на основе тестовых кейсов стресс тестирования.
5. Тестирование белого ящика – тестирование программного обеспечения, направленное на тестирование компонент системы, например, алгоритмов внутреннего устройства.
6. Юзабилити-тестирование – тестирование программного обеспечения, направленное на проверку удобности ли программный продукт конечному потребителю для применения.

Этапы тестирования рассмотрены в таблице 5.1.

Так же существует множество других видов и подвидов тестирования таких как тестирование чёрного ящика, тестирование локализации, тестирование безопасности, тестирование интерфейса пользователя, тестирование совместимости и т.д.

В данном дипломном проекте ограниченно функциональным тестированием программного обеспечения. Функциональное тестирование условно можно разделить на критическое и расширенное.

Критическое тестирование – поиск ошибок программы при её нормальной (правильной) работе. Проверка функций приложения на предмет правильности работы, а также проверка основной массы требований к продукту. Для данного вида тестирования пишутся наиболее подробные и глубокие тест-кейсы, чтобы покрыть всю возможную функциональность приложения.

Расширенное тестирование – это процесс поиска ошибок программы при нестандартных ситуация, например, неправильное заполнение полей email-адреса, переполнение массивов данных, ввод специальных символов, проверка граничных значений и т.д.

Таблица 5.1 – Этапы тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Вид тестирования | Этап тестирования | Объект тестирования | Критерий тестирования |
| Структурное, надежности | Разработка | Компоненты | Покрытие ветвлений, функций |
| Сборочное | Разработка | Подсистемы | Функциональность, степень проверки компонентов |
| Функциональное | Разработка | Система в целом | Соответствие функциональным требованиям ТЗ |
| Регрессионное | Разработка, сопровождение | Система в целом | Проверка качества внесения изменений |
| Нагрузочное | Разработка, сопровождение | Система в целом | Оценка характеристик системы, подбор конфигурации оборудования в соответствии с ТЗ |
| Стрессовое | Разработка, сопровождение | Система в целом | Корректность работы системы при пиковых нагрузках |
| Юзабилити-тестирование | Разработка, сопровождение | Система в целом | Удобство использование ПО пользователем |

## **5.1** Веб-приложение

Тестирование веб-приложения проводилось в многочисленных версиях пяти основных браузеров:

* Internet Explorer;
* Google Chrome;
* Opera;
* Mozilla Firefox.

В таблице 5.2 приведены примеры тестовых случаев для критического функционального тестирования: просмотр, добавление и удаление данных веб-приложения.

Таблица 5.2 – Критические тестовые случаи для веб-приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название теста | Описание тестового случая | Ожидаемые результаты | Тест пройден |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Тест сохранения изображения | 1. Запустить веб-приложение.  2. Нажать на кнопку «Sign in».  3. Ввести адрес электронной почты и пароль.  4. Нажать на кнопку «Login».  5. Нажать на кнопку «New image».  6. Заполнить все поля. Выбрать изображение для загрузки.  7. Нажать кнопку «Submit» | 1. Приложение запускается и отображается домашняя страница приложения.  2. Отображается форма авторизации пользователя с полями для ввода электронного адреса и пароля.  3. Все поля заполнены, становится доступной кнопка «Login».  4. Авторизация происходит успешно, пользователь перенаправляется на страницу с изображениями.  5. На экране появляется формы для заполнения идентификационного номера пользователя и названия изображения, а также поля для выбора загружаемого изображения.  6. Все поля заполнены, становится доступной кнопка «Submit».  7. Происходит возвращение к таблице со всеми изображениями, теперь там присутствует загруженное изображение. | Да |

*Продолжение таблицы 5.2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название теста | Описание тестового случая | Ожидаемые результаты | Тест пройден |
| 1 | 2 | 3 | 4 | 5 |
| 2 | Тест удаления изображения | 1. Запустить веб-приложение. 2. Нажать на кнопку «Sign in».  3. Ввести адрес электронной почты и пароль.  4. Нажать на кнопку «Login».  5. Нажать на кнопку «Destroy» рядом с тестируемым изображением. | 1. Приложение запускается и отображается домашняя страница приложения.  2. Отображается форма авторизации пользователя с полями для ввода электронного адреса и пароля.  3. Все поля заполнены, становится доступной кнопка «Login».  4. Авторизация происходит успешно, пользователь перенаправляется на страницу с изображениями.  5. Из таблицы удаляется тестируемое изображение со всей информацией. | Да |
| 3 | Тест авторизации пользователя | 1. Запустить веб-приложение.  2. Нажать на кнопку «Sign in».  3. Ввести адрес электронной почты и пароль.  4. Нажать на кнопку «Login». | 1. Приложение запускается и отображается домашняя страница приложения.  2. Отображается форма авторизации пользователя с полями для ввода электронного адреса и пароля.  3. Все поля заполнены, становится доступной кнопка «Login».  4. Авторизация происходит успешно, пользователь перенаправляется на страницу с изображениями. | Да |

*Продолжение таблицы 5.2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 4 | Тест создания пользователя | 1. Запустить веб-приложение.  2. Нажать на кнопку «Sign up».  3. Ввести адрес электронной почты и пароль.  4. Нажать на кнопку «Create». | 1. Приложение запускается и отображается домашняя страница приложения.  2. Отображается форма создания пользователя с полями для ввода электронного адреса и пароля.  3. Все поля заполнены, становится доступной кнопка «Create».  4. Создание происходит успешно, пользователь перенаправляется на страницу с изображениями. | Да |

В таблице 5.3 приведены примеры тестовых случаев для углубленного функционального тестирования нестандартных случаев.

Таблица 5.3 – Углубленные тестовые случаи для веб-приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название теста | Описание тестового случая | Ожидаемые результаты | Тест пройден |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Авторизация несуществующего пользователя | 1. Запустить веб-приложение.  2. Нажать на кнопку «Sign in».  3. Ввести неверный адрес электронной почты и пароль.  4. Нажать на кнопку «Login». | 1. Приложение запускается и отображается домашняя страница приложения.  2. Отображается форма авторизации пользователя с полями для ввода электронного адреса и пароля.  3. Все поля заполнены, становится доступной кнопка «Login».  4. Авторизация пользователя не происходит. Отображается сообщение «Неверный логин или пароль» | Да |

*Продолжение таблицы 5.3*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 2 | Пустая таблица изображений | 1. Запустить веб-приложение.  2. Нажать на кнопку «Sign in».  3. Ввести неверный адрес электронной почты и пароль.  4. Нажать на кнопку «Login». | 1. Приложение запускается и отображается домашняя страница приложения.  2. Отображается форма авторизации пользователя с полями для ввода электронного адреса и пароля.  3. Все поля заполнены, становится доступной кнопка «Login».  4. Авторизация происходит успешно. Пользователь видит сообщение «Загруженных изображений нет» | Да |

## **5.2** Управляющая программа

В таблице 5.4 приведены примеры тестовых случаев для критического функционального тестирования управляющей программы: создание фотографии, отправка POST запроса.

Таблица 5.4 – Критические тестовые случаи для управляющей программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название теста | Описание | Ожидаемые результаты | Тест пройден |
| 1 | Создание фотографии. | 1. Запустить управляющую программу.  2. Открыть фотографию с названием, заданным в константе FILENAME. | 1. Программа запускает, создает фотографию, делает запрос и заканчивает работу.  2. Отображается созданная фотография.  . | Да |
| 2 | Отправка POST запроса | 1. Запустить управляющую программу.  2. Открыть фотографию с названием, заданным в константе FILENAME. | 1. Программа запускает, создает фотографию, делает запрос и заканчивает работу.  2. На сервер веб-приложения приходит POST запрос с созданной фотографией | Да |

Как видно из таблиц, веб-приложение и управляющая программа справляются с тестами, что говорит об их высокой работоспособности.