|  |  |  |
| --- | --- | --- |
| **КП10. Классификация тестов по вашему приложению (вб)** | | |
| **1** | **Различные виды тестирования** | |
| 1.1 | Функциональные тесты  (Functional testing) | Проверяем основную функциональность приложения:   1. Проверяем поле регистрации. Вводим корректные данные для регистрации. Проверяем, что регистрация проходит успешно. 2. Проверяем поле поиск. Проверяем, что поле поиск работает по валидным значениям, ищет по введенным значениям, и не ищет по не указанным. 3. Проверяем работу фильтров и сортировки на релевантность выдачи. Выбираем товар, устанавливаем фильтры. Проверяем, что сортировка по фильтрам отображает товар, соответствующий выбранным фильтрам, выборка релевантная. |
| 1.2 | Тесты производительности (Performance testing) | Определяем границы приемлемой производительности:  - загрузка страницы 0,1-0,2 сек;  - ответ\ обратная связь от действия пользователя (загрузка фото, ввод данных) – 0,5-1,0 сек.  Проверяем изменение производительности:  - при увеличении нагрузки;  - при увеличении интенсивности выполнения операций.  При отклонении от заданных параметров более чем на 20% принимаем решение об оптимизации производительности. |
| 1.3 | Нагрузочные тесты  (Load testing) | Нагрузить функции приложения до предела, чтобы  узнать до какой нагрузки сервис остается функционален. Например:   1. Настроить Apache JMeter (или другой сервис) для проверки одновременной работы максимально возможного количества пользователей.   - 10 запросов одновременно. Скорость ответа (с). % ошибок  - 100 запросов одновременно. Скорость ответа (с). % ошибок  - 10 000 запросов одновременно. Скорость ответа (с). % ошибок  - 100 000 запросов одновременно. Скорость ответа (с). % ошибок  Проверяем при какой максимальной нагрузке приложение работает без потери скорости, и ошибок. |
| 1.4 | Тестирование надёжности  (Reliability testing) | Проведем проверку работоспособности нашего приложения при длительном (многочасовом, например поставим сервер на неделю) тестировании со средним уровнем нагрузки (расчетная нагрузка по ТЗ+%).  Проверим с самой обычной средней нагрузкой, не будет ли падений, неожиданного поведения, вызванного утечками памяти, переполнения базы, потери памяти.  Т.е. проверяем стабильность работы приложения. |
| 1.5 | Тестирование совместимости (Compatibility testing) | 1. Проверить запуск и работу приложения вб в разных браузерах: Chrome, Firefox, Opera, Edge, Safari 2. Проверить запуск и работу приложения вб на разных ОС: Windows (разные версии), IOS (разные версии) 3. Проверить запуск и работу приложения вб на разных устройствах. Использовать разные модели и конфигурации.   Стационарный компьютер, Laptop, мобильный телефон, планшет.   1. Проверить запуск и работу приложения вб с разных сетевых устройств, открыть на нескольких мониторах, разного разрешения. |
| **2** | **Различные типы тестов** | |
| 2.1 | Позитивные  (укажите и функциональные и нефункциональные) тесты | 1. Проверка строки поиск по валидным значениям: по слову, по подсказке, по фото, разными способами ввода, ввод символов, регистрозависимость, табуляция. 2. Проверка регистрации с нормальным именем: ввод логина, пароля, получения смс. 3. Выбор адреса и способа доставки из валидных значений |
| 1. Удобство использования приложения. Оценить интуитивность использования и графический дизайн. 2. Проверить нет ли наложение текста в баннерах и сообщениях, выход за рамки области 3. Защищенность. Войти в веб-приложение, используя действительные учетные данные. Выйти из приложения. Нажать кнопку НАЗАД браузера. Убедится, что вас попросили снова войти в систему или вы можете вернуться на страницу входа снова. |
| 2.2 | Негативные  (укажите и функциональные и нефункциональные) тесты | 1. Ввод недопустимых значений в поле регистрации 2. Ввод максимально больших значений в строку поиска (более 300 символов) 3. Загрузка недопустимых форматов файлов в поле поиск (более 8мб, pdf, csv, docx) |
| 1. Нагрузка. Проверить возможность работы приложения при одновременной работе максимально возможного количества пользователей. 2. Производительность. Проверить скорость загрузки фото, с максимально допустимым размером 3. Защищенность. Тестирование приложения к различному виду инъекций (SQL, SOAP, LDAP, XPATH и т.д.) |
| **3** | **Различные форматы тестирования** | |
| 3.1 | Скриптовое тестирование (по тестам): для каких (конкретно) частей продукта и в каких случаях будет полезнее всего? | Применяется в случаях:  - При работе над большим проектом/ в большой команде  - когда есть возможность планировать задачи, есть доступ к ТЗ и присутствует много документации  - сложный продукт, много неочевидных моментов  - достаточно времени и бюджета на проект  Для тестирования бизнес-правил, безопасности. Тестирование функциональных полей приложения: поле ввода, поле регистрации, функции оплаты, функция сортировки и фильтра. |
| 3.2 | Исследовательское тестирование (свободным поиском): для каких частей продукта и в каких случаях будет полезнее всего? | Применяется в случаях:  - когда нет доступа к ТЗ, нет возможности задать вопрос заказчику\разработчику  - постоянный коллектив,  - сотрудники с высокой квалификацией  - чтобы обеспечить быструю обратную связь о новом продукте,  - когда нужно быстро изучить продукт,  - небольшой проект с маленькой командой.  Для тестирования нашего приложения, не имея ТЗ, мы можем использовать туры Джеймса Уиттакера.  Например, проведем «Тур супермодели», проверим *Юзабилити и Интерфейс*, узнаем о внешнем интерфейсе приложения: красив ли, привлекателен ли, правильно ли используются цвета, нет ли лишних артефактов в интерфейсе, нет ошибок и опечаток в названиях и баннерах, соответствует ли интерфейс принятым стандартам (сравниваем с конкурентами), ожиданиям пользователя.  Так же полезным будет «Тур Коллекционера». Составим маинд карту приложения, и шаг за шагом будем проходить по всему *функционалу* приложения. Таким образом сможем протестировать все функциональные возможности приложения, чтобы примерно оценить тестовое покрытие, а также необходимость дополнительных исследований. |
| 3.3 | Автоматизированное тестирование: для проверки какого функционала наиболее полезно? Почему именно для него? | Применяется, когда тест подразумевается выполнение большого числа повторяющихся задач, содержит много вычислений и комбинаций.  При *регрессионном тестировании*, так как тестировщики теряют концентрацию и начинают пропускать дефекты.  Для проведения *нагрузочного* тестирования и *производительности*. Позволяет учесть все возможные комбинации взаимодействия пользователя с интерфейсом в веб-браузерах и прочих системах.  Например:  - тестирование «поля поиск» на производительность, при одновременном использовании несколькими пользователями, замеряем скорость работы поиска.  - на работу функции «скидка», когда нам надо перебрать и проверить большое количество вариантов и комбинаций. |