1. Опишите общий процесс тестирования отдельного модуля SPA приложения.

Тестирование отдельного модуля одностраничного приложения (SPA) может быть разделено на несколько этапов:

a. Подготовка: Разработка тестового плана и определение требований к тестированию. Определение основных функций модуля, которые будут протестированы.

b. Юнит-тестирование: Тестирование отдельных функций и компонентов модуля на предмет корректной работы. Обычно для этого используются специальные инструменты, такие как Jest или Mocha.

c. Интеграционное тестирование: Тестирование взаимодействия модуля с другими частями приложения и проверка корректности работы интеграции. Это может включать проверку API-запросов, корректное отображение данных и другие аспекты.

d. E2E (end-to-end) тестирование: Тестирование полного процесса работы модуля с точки зрения пользователя. Этот этап включает тестирование пользовательского интерфейса и проверку того, что пользователь может успешно выполнять задачи с использованием модуля. Для этого обычно используются инструменты типа Cypress или Selenium.

e. Регрессионное тестирование: Проверка того, что внесенные изменения или исправления не вызвали нежелательных побочных эффектов в других частях приложения.Баг-репорт должен иметь:

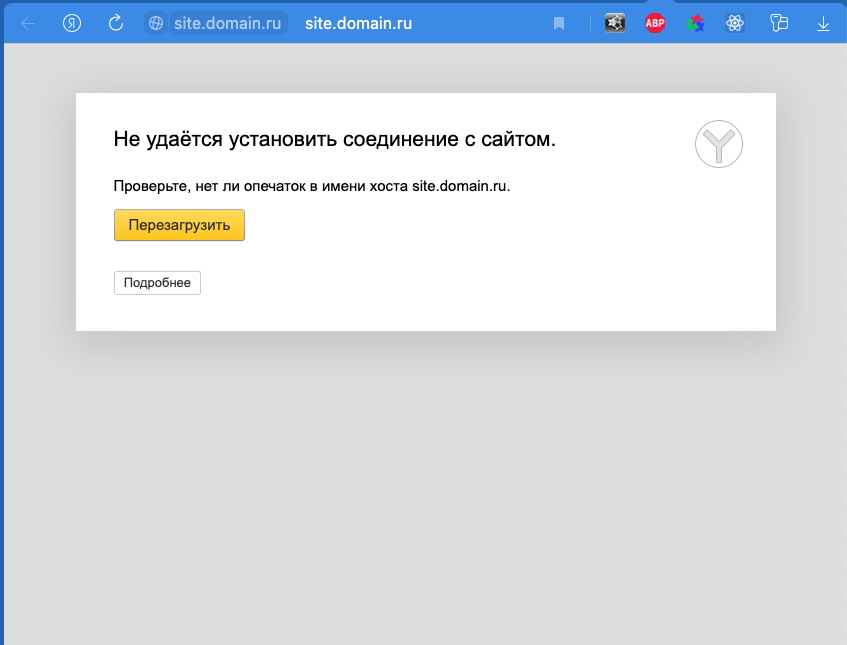
Id, приоритет, серьезность, заголовок, предварительные действия (опционально), шаги для воспроизведения бага, ожидаемый результат, фактический результат, постдействия(опционально). При необходимости к нему нужно прикрепить скриншоты или скринкасты. Так же необходимо описать окружение.



|  |  |
| --- | --- |
| id 00001 | |
| priority | Higer |
| severity | Blocked |
| При включении чайника, модели "NNN" с водой, не происходит нагрев воды | |
| 1. Подставку чайника включить в сеть питания 220В | |
| 2. Чайник наполнить холодной водой до отметки, указанной в инструкции к чайнику | |
| 3. Установить чайник на подставку | |
| 4. Нажать на чайнике кнопку включения | |
| ОР: На чайнике загорелся световой индикатор работы. Через 2 минуты корпус чайника стал теплым. | |
| ФР: Световой индикатор не горит. Корпус чайника остается холодным | |
| Окружение: Бытовая сеть 220В. Городской водопровод | |



|  |  |
| --- | --- |
| id 00002 | |
| priority | Higer |
| severity | Blocked |
| При входе на сайт https://site.domain.ru не удается получить соединение с сайтом | |
| 1. Зайти браузере на сайт https://site.domain.ru | |
| ОР Произошел вход на главную страницу сайта | |
| ФР. Браузеру не удается получить соединение с сайтом | |
| Окружение. MacOS 12.6.3 Monterey, Google Chrome Версия 111.0.5563.64 | |



1. Тестирование регистрации нового пользователя в приложении.

Сначала нужно ознакомиться с документацией, понять как должен проходить процесс регистрации их каких этапов он состоит? Какие передаваемые данные считаются валидными?

Предположим, что будет простая регистрация пользователя. Три поля для ввода данных.

- Поле «Имя» принимает кириллицу, латиницу в двух регистрах. Иероглифы и другие алфавиты, а так же дефис и пробел. До 50 символов

- Поле «Email». Только латиница в двух регистрах, цифры, спецсимволы.

- Поле «Пароль». Цифры, латиница в двух регистрах, символы дефиса и нижнего подчеркивания. Не менее 5 и не более 15 символов. Пароль обязательно должен содержать хотя бы одну букву любого регистра или хотя бы одну цифру. Существует проверка уровня надежности пароля. Разрешен ввод пароля копипастом

- Кнопка «Зарегистрироваться»

Сначала нужно провести серию позитивных проверок. Проверить при помощи техники классов эквивалентности поля на ввод разных допустимых значений.

В поле имя: Самое короткое, среднее и самое длинное имя. Имена на кириллице латинице и иероглифах. Имена с буквами в верхнем и нижнем регистре. Двойные имена, содержащие пробел, содержащие дефис. Состоящие более чем из двух слов.

Проверить отправку письма подтверждения регистрации на Имейл пользователя.

Потом провести серию негативных проверок.

Попытка регистрации с пустым полем «Имя»

Попытка регистрации с пустым полем «Имейл»

Попытка регистрации с пустым полем «Пароль»

Попытка регистрации с недопустимыми символами в поле «Имя»

Попытка регистрации со строкой недопустимой длины в поле «Имя»

Попытка регистрации с недопустимыми символами в поле «Имейл»

Попытка регистрации с недопустимыми символами в поле «Пароль»

1. Для создания новой сущности нужно использовать POST запрос. Узнать из Сваггера или, если его нет, у разработчика как оформить запрос. URL, заголовки, содержание тела запроса, получение прав доступа (при необходимости). Отправить запрос. Получить ответ. Из сваггера или от разработчика узнать каков должен быть ожидаемый ответ и статус и сравнить его с фактическим. Если запрос выполнен успешно, то нужно сделать SQL запрос в базу данных и запросить только что созданную информацию. Если запрос из БД совпадает с запросом API, то можно утверждать, что метод POST работает исправно.

Условный пример:

тело POST запроса.

{

“author”: “Достоевский Ф. М.”,

“title”: ”Записки из мертвого дома”,

“price”: 420,

“amount”: 50

}

Тело ответа:

{

“id”: 0001,

“author”: “Достоевский Ф. М.”,

“title”: ”Записки из мертвого дома”,

“price”: 420,

“amount”: 50

}

Запрос в БД:

SELECT “author”, “title”, “price”, “amount”

FROM table\_books

WHERE “id” = 0001;

|  |  |
| --- | --- |
| id 00003 | |
| priority | Higer |
| severity | Critical |
| Страница авторизации позволяет осуществить вход в систему с пустым полем «username» | |
| Предварительные действия. На сайте https://\*\*\* должен быть зарегистрирован аккаунт | |
| 1. Зайти браузере на сайт https://\*\*\* | |
| 2. Нажать на кнопку авторизации | |
| 3. Ввести существую корректный пароль для входа в аккаунт | |
| 4. Поле «username» оставить незаполненным | |
| 5. Нажать а кнопку "Войти" | |
| ОР Кнопка "Войти" остается неактивной. Вход в аккаунт невозможен | |
| ФР. Произошел вход в аккаунт пользователя | |
| Окружение. MacOS 12.6.3 Monterey, Google Chrome Версия 111.0.5563.64 | |

1. Передавать бинарные значения в JSON нельзя. Формат JSON изначально не поддерживает двоичные данные. Бинарные данные должны быть экранированы так, чтобы их можно было помещать в строковый элемент (т.е. Ноль или несколько символов Unicode в двойных кавычках с использованием обратных слэшей) в JSON
2. Тестовые наборы данных для поля ввода даты, которое отсеивает пользователей в возрасте до 18 лет.

**Негативные проверки**

Дата рождения = Текущая дата

Месяц = текущий месяц

Год рождения > текущий год – 18

Дата рождения = Текущая дата

Месяц > текущий месяц

Год рождения = текущий год – 18

Дата рождения > Текущая дата

Месяц = текущий месяц

Год рождения = текущий год – 18

Месяц < текущий месяц

Год рождения > текущий год – 18

**Позитивные проверки**

Дата рождения = Текущая дата

Месяц = текущий месяц

Год рождения = текущий год – 18

Дата рождения < Текущая дата

Месяц = текущий месяц

Год рождения = текущий год – 18

Месяц < текущий месяц

Год рождения = текущий год – 18

Год рождения < текущий год – 18

10. Как протестировать скачивание файла при недостаточном количестве места на диске.

Для тестирования скачивания файла при недостаточном количестве места на диске можно использовать виртуальную машину или контейнер с ограниченным размером диска или эмуляцию этой ситуации. Вот шаги для тестирования:

a. Настройка тестового окружения: Создайте виртуальную машину или контейнер с ограниченным пространством на диске или настройте систему так, чтобы имитировать недостаток места на диске. Убедитесь, что доступное место меньше, чем размер файла, который вы планируете скачать.

b. Выполнение теста: Используя приложение, попытайтесь скачать файл достаточного размера, чтобы превысить доступное пространство на диске. Обратите внимание на поведение приложения в процессе скачивания.

c. Проверка результата: После попытки скачать файл убедитесь, что приложение корректно обрабатывает ситуацию с недостаточным местом на диске. Оно должно предоставлять соответствующие предупреждения или ошибки, информируя пользователя о проблеме. Возможные реакции приложения могут включать:

Остановка скачивания с выводом сообщения об ошибке или предупреждения для пользователя.

Предложение пользователю освободить место на диске, удалив ненужные файлы или перемещая их в другое место.

Возможность продолжить скачивание на другой диск или в другое место с достаточным свободным пространством.

Проверьте, что приложение обрабатывает эту ситуацию корректно и предоставляет пользователю понятные инструкции для разрешения проблемы.

11. Как протестировать загрузку файла при недостаточном количестве места на сервере.

Для тестирования загрузки файла при недостаточном количестве места на сервере можно использовать тестовый сервер с ограниченным пространством на диске или с эмулированным недостатком места. Затем попытаться загрузить файл и проверить, что сервер возвращает соответствующую ошибку или предупреждение. Вот шаги для тестирования:

a. Настройка тестового сервера: Установите сервер с ограниченным пространством на диске или настройте сервер так, чтобы он эмулировал недостаток места (например, настройка сервера для возврата ошибки 507 Insufficient Storage).

b. Выполнение теста: Попытайтесь загрузить файл на сервер через приложение. Убедитесь, что вы выбрали файл достаточного размера для воспроизведения ситуации с недостаточным местом на сервере.

c. Проверка результата: После попытки загрузки файла убедитесь, что сервер возвращает ожидаемую ошибку (например, код ошибки 507) и что приложение корректно обрабатывает эту ошибку, показывая пользователю соответствующее сообщение или предупреждение.

12. Примеры отрицательных тест-кейсов для проверки веб-страница с полями: e-mail, password и кнопкой submit

- поле «e-mail» - заполнено согласно требований

- поле «password» - не заполнено

- поле «e-mail» - не заполнено

- поле «password» - заполнено согласно требований

- поле «e-mail» - Превышение длины локальной части (максимальная допустимая 64 символа)

- поле «password» - заполнено согласно требований

- поле «e-mail» - Превышение длины доменного имени (максимальная допустимая 255 символов)

- поле «password» - заполнено согласно требований

- поле «e-mail» - Превышение длины участка доменного имени между точками (максимальная допустимая 63 символа)

- поле «password» - заполнено согласно требований

- поле «e-mail» - Отсутствие @ в email

- поле «password» - заполнено согласно требований

- поле «e-mail» - Отсутствие локальной части

- поле «password» - заполнено согласно требований

- поле «e-mail» - Отсутствие доменной части

- поле «password» - заполнено согласно требований

- поле «e-mail» - Содержит две точки подряд

- поле «password» - заполнено согласно требований

- поле «e-mail» - Локальная часть начинается с . (точки)

- поле «password» - заполнено согласно требований

- поле «e-mail» - Доменная часть начинается с . (точки)

- поле «password» - заполнено согласно требований

- поле «e-mail» - Содержит один или более пробел между символами

- поле «password» - заполнено согласно требований

- поле «e-mail» - заполнено согласно требований

- поле «password» - короче минимально допустимого

- поле «e-mail» - заполнено согласно требований

- поле «password» - содержит неразрешенные спецсимволы

- поле «e-mail» - заполнено согласно требований

- поле «password» - содержит неразрешенные алфавиты

- поле «e-mail» - заполнено согласно требований

- поле «password» - содержит пробел между символами

- поле «e-mail» - заполнено согласно требований

- поле «password» - состоит из пробелов

13. Для тестирования работы функционала нужно, естественно, использовать функциональное тестирование.

Можно провести автоматизированное тестирование. Я бы мог использовать Python + Selenium + Pytest

Можно провести ручное тестирование используя технику классов эквивалентности. А так же технику граничных значений.

Тест-кейсы нужно разбить на две группы :валидные и невалидные.

В Валидные ввести кейсы с группами входных данных, которые допустимы требованиями для использования.

Например такие группы:

Числа с запятой

Числитель: 0,1

Знаменатель: 0,1

Числитель: 99,9

Знаменатель: 0,1

Числитель: 99,9

Знаменатель: 99,9

Числитель: 0,1

Знаменатель: 99,9

Числитель: 99,9

Знаменатель: 0,1

Числитель: 98,999999999 (максимально допустимое число знаков после запятой)

Знаменатель: 0,11111111 (максимально допустимое число знаков после запятой)

Целые числа

Числитель: 1

Знаменатель: 1

Числитель: 90

Знаменатель: 90

Числитель: 50

Знаменатель: 10

Числитель: 2

Знаменатель: 75

При этом нужно следить, сколько знаков после запятой выдает функционал, совпадает ли это с требованиями к продукту. Как происходит округление дробей.

Группы данных для невалидных тест-кейсов.

Числитель: не заполнен

Знаменатель: 10 (произвольное валидное число)

Числитель: 0

Знаменатель: 10 (произвольное валидное число)

Числитель: 100

Знаменатель: 10 (произвольное валидное число)

Числитель: Буквы

Знаменатель: 10 (произвольное валидное число)

Числитель: спецсимволы

Знаменатель: 10 (произвольное валидное число)

Числитель: пробелы

Знаменатель: 10 (произвольное валидное число)

Числитель: 10 (произвольное валидное число)

Знаменатель: не заполнен

Числитель: 10 (произвольное валидное число)

Знаменатель: 0

Числитель: 10 (произвольное валидное число)

Знаменатель: 100

Числитель: 10 (произвольное валидное число)

Знаменатель: Буквы

Числитель: 10 (произвольное валидное число)

Знаменатель: спецсимволы

Числитель: 10 (произвольное валидное число)

Знаменатель: пробелы