

ТЕСТОВАЯ ДОКУМЕНТАЦИЯ

Тест план — это документ, описывающий весь объем работ по тестированию, начиная с описания объекта тестирования, стратегии, расписания, критериев начала и окончания тестирования. (Тест-план пишет тим-лид).

Тест-план отвечает на следующие вопросы:

1. Что необходимо тестировать?

- Описание объекта тестирования: системы, приложения.

2. Что будем тестировать?

- Список функций и описание тестируемой системы или её компонентов в отдельности (форма регистрации, форма авторизации, чат, комментариев, лайк).

3. Как будем тестировать?

- Указываем те виды тестирования, которые будем использовать при тестировании.

4. Когда будем тестировать?

- Последовательность проведения работ: чтение технической документации(требований), создание тестовой документации, тестирование, анализ результатов тестирования, заведение баг-репортов.

5. Критерии начала тестирования.

- Готовность тестового стенда (сайта, на котором мы тестируем), наличие всей необходимой документации, законченность разработки требуемого функционала.

6. Критерии окончания тестирования.

- Весь функционал работает согласно заявленным требованиям.

Чек-лист — это документ, описывающий что должно быть протестировано.

Как правило, чек-лист содержит только действия (шаги) т.е. то, что будем проверять (т.е. чек-лист — это список необходимых проверок).

Например, есть форма авторизации:

Авторизация



Электронная почта

Пароль

Напомнить пароль

Войти

[Я в первый раз](#)

В чек-листе мы укажем что проверим в этой форме:

- Иконку закрытия формы
- Поле электронная почта
- Поле пароль
- Ссылку "Напомнить пароль"
- Кнопку "Войти"
- Ссылку "Я в первый раз"

Тест-кейс (тестовый сценарий) — это документ, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Тест-кейс состоит из 3-х частей:

Предварительные условия - что нужно сделать до проведения тестирования, например, протестировать только на устройстве Android, или только в браузере Safari. (Предварительные условия не являются обязательной частью).

Шаги (Действия) — последовательность действий, которые мы должны осуществить для проверки тестируемого функционала с целью убедиться, что всё работает согласно заявленным требованиям.

Ожидаемый результат - что должно получиться по результатам наших действий.

Виды Тестовых Сценариев:

Тест-кейсы разделяются по ожидаемому результату на позитивные и негативные:

- **Позитивный тест-кейс** использует только корректные данные и проверяет, что приложение работает согласно заявленным требованиям.
- **Негативный тест-кейс** оперирует как корректными так и некорректными данными (минимум 1 некорректный параметр) и проверяет, что вызываемая приложением функция не выполняется. (т.е. на примере формы авторизации vk.com когда мы вводим корректный логин и некорректный пароль нам выведется уведомление о том, что введены некорректные данные, т.е. осуществились проверки полей и сработал обработчик ошибок, который вывел уведомление о неправильном пароле).

Сначала мы пишем позитивные тест-кейсы, чтобы убедиться, что функционал работает согласно заявленным требованиям, а затем создаём негативные тест-кейсы для того, чтобы убедиться в том, что, когда мы сделали что-то неправильно или ввели некорректные значения, у нас сработают обработчики ошибок и функция не выполнится.

Авторизация



[Напомнить пароль](#)

Войти

[Я в первый раз](#)

Тест-кейс № 1. Проверка функционала формы авторизации.

Шаги:

- Зайти на сайт zdorovr.ru.
- В шапке сайта в правом верхнем углу нажать на кнопку "Войти".
- В появившейся форме авторизации в поле "Электронная почта" ввести значение – sramotnik1@mail.ru.
- В поле "Пароль" ввести значение – Sramota1.
- Нажать на кнопку "Войти".

Ожидаемый результат:

Отправится запрос на сервер, после которого в меню сайта появится пункт "Личный кабинет", при нажатии на который осуществится переход в профиль пользователя.

Тест-кейс № 2. Проверка формы авторизации без ввода логина и пароля.

Шаги:

- Зайти на сайт zdorovr.ru.
- В шапке сайта в правом верхнем углу нажать на кнопку "Войти".
- Оставить пустыми поля логин и пароль
- Нажать на кнопку "Войти".

Ожидаемый результат:

Поля "Электронная почта" и "Пароль" подсвелятся красным цветом с подсказкой, в которой будет указано, что поля являются обязательными для заполнения.

Отличия чек-листов от тест-кейсов

Чек-лист менее детализирован чем тестовый сценарий. Его уместно использовать тогда, когда тестовые сценарии будут избыточны. (Например, проверить функционал лайка, кол-ва просмотров записи на стене, функционал репоста).

Баг Репорт — это документ, описывающий ситуацию или последовательность действий, приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

The screenshot shows a Jira interface with a bug report. The top navigation bar includes 'Jira', 'Ваша работа', 'Проекты', 'Фильтры', 'Дашборды', 'Люди', 'Приложения', and a 'Создать' button. A search bar and user avatars are on the right. The breadcrumb trail is 'Проекты / TestingSchool / Добавить Эпик / TEST-34'. The bug title is 'Не работает кнопка "Войти" в форме авторизации'. Below the title are buttons for 'Прикрепить', 'Добавить дочернюю задачу', and 'Добавить ссылку на задачу'. The 'Описание' section contains a 5-step reproduction guide for a login issue on zdorovr.ru, the expected result (successful login), and the actual result (nothing happens). The 'Вложения' section shows a screenshot of the login form. On the right, a sidebar shows the bug's status as 'В работе' and a 'Сведения' section with fields for Author (Nikolay), Assignee (Андрей Назаров), Severity (Blocker), Priority (High), and Time Spent (Not registered). It also shows creation and update timestamps and a 'Настроить' button.

Проекты / TestingSchool / Добавить Эпик / TEST-34

Не работает кнопка "Войти" в форме авторизации

Прикрепить Добавить дочернюю задачу Добавить ссылку на задачу

Описание

1. Перейти на сайт zdorovr.ru
2. В шапке сайта в правом верхнем углу нажать на кнопку "Войти".
3. В появившейся форме авторизации в поле "Электронная почта" ввести значение – sramotnik1@mail.ru.
4. В поле "Пароль" ввести значение – [Sramota1](#).
5. Нажать на кнопку "Войти".

Ожидаемый результат: Отправится запрос на сервер, после которого в меню сайта появится пункт "Личный кабинет", при нажатии на который осуществится переход в профиль пользователя.

Фактический результат: При нажатии на кнопку "Войти" ничего не происходит.

Вложения ((1))

image-2021-08-... 212.png
05 авг. 2021, 00:46 AM

В работе

Сведения

Автор	Nikolay
Исполнитель	Андрей Назаров
Серьёзность	Blocker
Приоритет	High
Учет времени	Время не зарегистрировано

Создано 1 минуту назад
Дата обновления 1 минуту назад

Настроить

Основные составляющие баг-репорта

(обязательно спросят на собеседовании, их интересует что ты скажешь про приоритет и серьёзность)

Короткое описание (Заголовок) - короткое описание проблемы, явно указывающее на причину и тип ошибочной ситуации. (Пишется кратко и лаконично. Т.е. что, где сломалось. Например, в форме авторизации не работает кнопка "Войти").

Описание - в описании указываем предварительные условия, шаги, ожидаемый и фактический результаты. (Ожидаемый результат — это то, что должны были получить по требованиям, фактический — это то, что получили по итогу, т.е. то что сделал разработчик.)

Проект - название тестируемого проекта (если вы работаете в команде, у которой много проектов, то нужно указывать в каком проекте произошла ошибка, если проект один, то он будет выбран по умолчанию).

Статус - статус бага. На какой стадии жизненного цикла бага находится дефект (смотри первый урок).

Автор - создатель баг репорта.

Исполнитель - сотрудник, назначенного на решение проблемы.

Прикрепленные файлы - скриншот или любой другой документ, который может помочь прояснить причину ошибки или указать на способ решения проблемы.

Серьезность (Severity) - влияние дефекта на работоспособность (т.е. насколько дефект влияет на работу приложения). Выставляется тестировщиком.

- Блокирующий (Blocker)
- Критический (Critical)
- Значительный (Major)
- Незначительный (Minor)
- Тривиальный (Trivial)

Приоритет (Priority) - очередность выполнения бага. Выставляется тимлидом отдела тестирования. Чем выше приоритет, тем быстрее нужно исправить дефект.

Приоритет дефекта:

- Высокий (High)
- Средний (Medium)
- Низкий (Low)

Градация Серьезности дефекта:

Блокирующая (Blocker)

Дефект полностью блокирует выполнение функционала, нет никакого способа его обойти. (Мы открыли форму авторизации, заполнили все поля, нажали на кнопку "Войти" и ничего не происходит. На сайте есть только одна форма авторизации и мы не можем другим никаким другим способом войти в профиль.)

Критическая (Critical)

Дефект блокирует часть функциональности, но есть альтернативный путь для его обхода. (Мы заполнили форму регистрации, нажали на кнопку "Зарегистрироваться" и ничего не происходит. Таким образом мы не можем зарегистрироваться, но есть кнопка "Зарегистрироваться через социальные сети", при нажатии на которую мы можем выбрать соцсеть, через которую можно создать профиль. Например, через Google-аккаунт, или через профиль в Instagram).

Значительная (Major)

Дефект, указывающий на некорректную работу части функциональности.

Существует более одной способа реализации функционала. Наиболее часто встречаются дефекты, которые можно отнести именно к этому уровню серьезности.

(Например, на главной странице сайта есть форма для получения кредита и она не работает, но заявку на кредит можно заполнить в личном кабинете, во вкладке "Кредиты", во вкладке "Особые предложения" и т.п.)

Незначительная (Minor)

Дефект, не относящийся к функциональности системы. Такая серьезность проставляется для дефектов, которые относятся к удобству использования или интерфейсу. (Текст выходит за рамки блока, слипающиеся кнопки, съехавшая иконка, картинка выезжает за пределы блока).

Тривиальная (Trivial)

Обычно это грамматические, орфографические и пунктуационные дефекты.

Градация Приоритета дефекта

Высокий

Требуется исправить в первую очередь.

Средний

Требуется исправить во вторую очередь, когда нет дефектов с высоким приоритетом.

Низкий

Исправляется в последнюю очередь, когда все дефекты с более высоким приоритетом уже исправлены.

Как правило, чем выше серьёзность, тем выше приоритет.

Пример дефекта с **низкой серьёзностью и высоким приоритетом**: ошибка в названии компании, неправильный логотип компании, непристойные материалы на сайте, ошибки на главной странице сайта.(Т.е. на работу сайта эти дефекты никак не влияют, но исправить их нужно немедленно).

Пример дефекта с **высокой серьёзностью, но низким приоритетом**: ошибки в функционале, который будет разработан и выпущен в будущем, но реализуется сейчас, например, компания разрабатывает программное обеспечение по бухгалтерскому учёту. Сейчас идёт 3-й квартал, но разработчик уже делает функционал годового отчёта, но в нём есть критические баги. Но так как годовой отчёт нужен будет только через полгода (т.к. годовую отчётность публикуют по итогам года, а сейчас идёт 3-й квартал), то проблема в этом функционале на данный момент времени не приоритетна.