

## Общее о тестировании приложения Treffly

**Цель тестирования:** обеспечение повышение качества мобильного приложения Treffly

**Основные фичи проекта, на которые больше всего будет уделено внимания:**  
авторизация/регистрация, поиск мероприятия, создание мероприятия

**Слабые места тестирования:** незнакомая технология PWA-приложение, корректная работа геолокации, особенно для поиска ближайших мероприятий (есть проблемы в Воронеже с гео)

**Техники:** ручное тестирование (компонентное , смоук, регресс)

**Инструмент ведения тестов:** гугл таблицы

## План тестирования приложения Treffly

### 1. Раннее подключение QA

QA начинает работу еще на этапе написания технического задания (ТЗ) и проектирования дизайна. На этом этапе:

- Проверяет ТЗ и макеты, выявляет неточности
- Задает уточняющие вопросы
- Предупреждает возможные ошибки до начала разработки

### 2. Процесс тестирования

- Разрабатывает тестовые сценарии
- Проводит тестирование продукта
- Документирует результаты

### 3. Проверка функциональности и работа с багами

- Выполняет тест-кейсы и чек-листы
- Проверяет соответствие продукта требованиям
- Идентифицирует и документирует ошибки
- Передает информацию разработчикам
- Проверяет исправления и закрывает баги

### 4. Актуализация тестовой документации (при необходимости)

- Обновляет тест-кейсы и чек-листы
- Фиксирует изменения в продукте

## Процесс тестирования новых фич

1. QA заранее пишет тест-кейсы для каждой фичи
2. Разработчики передают QA сборку PWA-приложения
3. Получив сборку, запускает тестирование и проверяет работу компонентов
4. Если обнаружены баги:
  - Определяет их приоритет
  - Передает информацию разработчикам
5. После исправления багов:
  - Проверяет их и закрывает
  - Если затронуты другие фичи, проводит регрессионное тестирование
6. В конце спринта желательно проводить smoke-тестирование ключевых бизнес-функций
7. После успешного тестирования QA подтверждает, что фичу можно влить в основную ветку

## Чек-лист проверки ТЗ

1. В ТЗ проекта обязательно должны быть указаны версии ОС и браузеров, поддерживаемые будущим приложением.
2. Для каждой из фич необходимо наличие описания всех переходов, отраженных в дизайне. Нужно предельно прозрачно понимать, откуда куда мы переходим, а также должна ли быть возможность вернуться обратно и каким образом(в случае неочевидных возвратов)
3. Необходимо определить для каждого экрана порядок загрузки данных, учитывать обязательность запросов, а также обработку успешных ответов и возможных ошибок при их выполнении
4. Способы обновления страницы - птр, кнопка, автообновление, должна ли обновляться ли информация после возврата с дочерних экрана.
5. Error states, Empty states, Not Found и их обработка - снэки, тосты, диалоги, алерты.
6. В случае наличия полей ввода, необходимо указание их максимальной мощности(количество символов), автофокуса, доступности различных символов для ввода

## Чек-лист написания проверок

1. Писать подробные шаги , что нужно сделать, куда нажать, что ввести
2. Если необходимо прилагать слепки дизайна
3. Прописывать подробный ожидаемый результат к каждому шагу в проверке, что должны увидеть, что должно произойти

## **Дополнительный чек-лист для проверки верстки**

1. Пресстейты есть у всех кликабельных элементов и они соответствуют дизайну
2. Области нажатия всех кликабельных элементов соответствуют стандарту - сложностей при взаимодействии с элементом нет
3. Навбар/тулбар заскралливается согласно дизайну
4. Отступы стрелок и крестиков в навбаре/тулбаре соответствуют дизайну
5. При птр анимация прогресс-лоадера отрабатывает без рывков, не двоится и длится столько, сколько идет запрос
6. Загрузка шиммером должна соответствовать дизайну
7. Переходы между экранами отрабатываются плавно
8. Тексты соответствуют макетам/документу с текстами