**Тест-план раздела лайки в вк**

**Цель тестирования:** проверить работу функционала раздела «лайки» в вк, для корректной работы раздела.

Объект тестирования: лайки в вк

Область тестирования:

1. Возможность ставить и снимать лайки под разными публикациями (фото, видео, истории, посты)
2. Количество лайков
3. Работа с ошибками (например, если человек не авторизован и ставит лайк)
4. Проверка функциональности на различных платформах (Android, IOS, веб)

Стратегии тестирования:

1. Техники тест-дизайна:

1.Эквивалентное разбиение

Разбиение на эквивалентные классы позволяет нам протестировать все, но при этом не проверяем каждую комбинацию.

Например: тестируем с разными типами публикаций (фото, пост, видео, истории).

2.Анализ граничных значений

Тестируем, что лайков 0, лайк 1. Также в день может быть достигнут лимит. Проверяем и границы лимита. На границах водятся баги, поэтому анализ граничных значений помогает выявить их.

3.Таблица принятия решений

Таблица принятия решений помогает «свежим» взглядом проанализировать и учесть все возможные условия, критерии и действия, при этом избегая пропусков.

4. Причина-следствие

Выявляем связи действиями пользователей и тем, что будет в результате

Например: поставили лайк, сердечко стало красным, пришло уведомление о новом лайке

5.Исследовательское тестирование

Тестировщики изучают функционал во время тестирования, чтобы была возможность выявить неожиданные проблемы.

1. Подходы к тестированию:

1.Ручное тестирование

Выявляем баги, которые заранее нельзя предусмотреть в автоматизации.

Тестируем, основной функционал раздела лайки.

2.Кросс-платформенное тестирование

Необходимо протестировать работу функционала лайков на разных платформах, чтобы не встретить баги из-за различных устройств.

3.Автоматизированное тестирование

С помощью автоматизации проверяем базовую функциональность лайков, например после релизов и обновлений.

Тестовые данные:

Тестовые аккаунты с разным доступом: зарегистрированные пользователи, заблокированные пользователи, пользователи с ограничением приватности

Тестовый контент: посты, фото, видео, истории

**Сценарии тестирования:**

**Функциональное:**

1)Позитивное

Нужно проверить, что функция лайков работает правильно.  
  
Тестируем, что мы можем ставить и снимать лайки на различные виды публикаций (фото, видео, посты, истории), получать уведомление о новых лайках, видим количество лайков.

Авторизированный пользователь может поставить лайк, видит количество лайков, неавторизированный пользователей не может поставить лайк, получаем уведомление о новых лайках.

2) Негативное

Нужно проверить, что при некорретных действиях пользователей будет происходить.   
  
Тестируем работу раздела лайки при неккоретных действиях или ошибках. Система будет реагировать как положено (корректно).

Пользователь пытается поставить лайк без интернета, тогда лайк не сохранится и бэкенд вернет ошибку. Пользователь пытается поставить много лайков на один и тот же пост, тогда лайк будет то ставится, то сниматься, но лайк будет один.

**Нефункциональное:**

3) Тестирование безопасности   
  
Нужно проверить, что лайки нельзя накрутить с использованием ботов, нельзя не авторизоваться и при этом поставить лайк.

Это и тестируем.  
  
4) Нагрузочное тестирование

Нужно проверить, будет ли работать все ок, если в одно время большое количество лайков будет ставить много пользователей.  
  
Тестируем устойчивость системы к нагрузкам.

5) Тестирование доступности   
  
Нужно протестировать, что функция лайков доступна для людей с ограниченными возможностями  
  
Тестируем доступность этих функций  
  
Например, что кнопку лайков можно поставить на клавиатуре.

6) Тестирование локализации  
  
Нужно проверить, что лайки корректно отображаются с разными настройками.  
  
Тестируем, что лайки отображаются при смене языка.  
  
При смене языка уведомление о лайках должно быть на другом языке, лайки отображаются в любом регионе, если его поменять в настройках.  
  
7)Smoke-тестирование   
  
Тестируем, что мы можем поставить лайк и убрать его, если вышла новая версия приложения.  
  
8) UX тестирование   
  
Нужно протестировать, что интерфейс для пользователя понятен, прост и удобен, то есть найти кнопку лайка просто, и что будет, если на нее нажать

9) UI тестирование

Тестируем, если нажать на сердечко, то оно станет красным, если нажать еще раз, то оно опять станет серым, анимация кнопки при нажатии работает плавно, без задержек.

Чек-лист UI тестирование:

1. Расположение кнопок, текста, изображений
2. Анимация лайков
3. Правильность текстов (кто поставил лайк), шрифты, заголовки, также это все в уведомлениях
4. Корректность работы анимаций и отображение интрерфейса в разных темах
5. Изменение элементов при изменении масштаба
6. Элементы не накладываются друг на друга в разных расширениях

10) Тестирование API

Когда ставим лайк, то отправляем запрос на бэкенд с соответствующими данными.

Также тестируем, что бэкенд возвращает одинаковые данные с разных устройств (при авторизации, лайк будет стоять на одном и том же, например посте, на разных устройствах)

Чек-лист API тестирование:

1. Возвращается корректный статус-код (пример, 200)
2. Обработка некорректных запросов
3. Данные синхронизируются между устройствами
4. Время откликов API при массовых запросах
5. Корректность обработки ошибок api на стороне сервера
6. Возвращение корректных данных для разных типов публикаций
7. Корректность запросов на бекенд
8. Логирование всех запросов и ошибок

9) Кросс-платформенное тестирование

Тестируем работу функционала на разных платформах: Android, IOS, MacOS, Windows, на разных размерах экранов, на разных версиях ОС.

Чек-лист Android/IOS:

1. Корректность Лайков в мобильном приложении
2. Смена ориентации экрана
3. Разные версии ios/android
4. Разные разрешения экранов
5. Свайпы страниц, уведомлений
6. Push-уведомления
7. Корректность отображения на девайсах с челкой и Dynamic Island
8. Корректность работы функционала при низком заряде батареи
9. Нестандартные оболочки Android (например, MIUI)

10) Кросс-браузерное тестирование

Тестируем функционал на разных браузерах Chrome, Safari, FireFox, и на разных версиях браузера, также на старых версиях браузера, если поддерживаются.

Чек-лист Web/Desktop:

1. Функционал на разных браузерах (Chrome, Safari, FireFox)
2. Работоспособность в разных версиях (старых, которые поддерживаются и более новых)
3. Совместимость с разными ОС (MacOS, Windows, Linux)
4. Работоспособность функционала при медленном интернете
5. Корректность работы уведомлений в разных браузерах
6. Разные версии браузеров
7. Режим инкогнито
8. Лайки с впн

Автоматизированное тестирование:

Тестируем отдельные блоки кода с помощью юнит-тестов, проверяем работу различных компонентов между собой, используем интеграционные тесты.

План ресурсов и сроков:

Ресурсы:

Тестовые данные:

Тестовые аккаунты с разным доступом: зарегистрированные пользователи, заблокированные пользователи, пользователи с ограничением приватности

Тестовый контент: посты, фото, видео, истории

Команда:

Кол-во тестировщиков

Например, 2 ручных тестировщика, 1 тестировщик по автоматизации.

Устройства для тестирования:

Mobile (Iphone 14, Samsung A52)

ПК (MacOS, Windows)

Кросс-браузерное тестирование:

Chrome

Safari

Firefox

Сроки (пример):

Анализ требований – 1 день

Написание тест-кейсов – 2 дня

Подготовка тестовой среды – 1 день

Ручное тестирование – 4 дня

Автотесты – 2 дня

Риски:

1. Неполные требования к функционалу
2. Не полный доступ к устройствам для тестирования
3. Задержки при тестировании в какой-либо среде

Предположения:

1. Разработчики быстро находят и фиксят имеющиеся баги
2. Доступ ко всем инструментам и документации, необходимых при тестировании будут вовремя доступны.

Критерии завершения тестирования:

1. Все критичные и блокирующие баги исправлены, функционал работает корректно, согласно требованиям разработчика и заказчика.
2. Кросс-платформенное, кросс-браузерное тестирование, бета-тестирование и регрессионное тестирование завершены с успехом.