

ПЛАН ТЕСТИРОВАНИЯ

Радел «Лайки» приложения ВКонтакте

Галина Татаринова

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1.1 ОСНОВНАЯ ИНФОРМАЦИЯ	3
1.2 ЦЕЛЬ	3
2. РАМКИ ПРОЕКТА	3
2.1 ОБЪЕКТ ТЕСТИРОВАНИЯ	3
3. ПЛАН РАБОТЫ	4
4. ПЛАН И СТРАТЕГИЯ ТЕСТИРОВАНИЯ	4
4.1 ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ	5
4.2 UI/UX ТЕСТИРОВАНИЕ	5
4.3 НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ	6
4.4 РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ	6
4.5 СООБЩЕНИЯ ОБ ОШИБКАХ	7
5. ТРЕБОВАНИЯ К ОКРУЖЕНИЮ	8
6. КРИТЕРИИ ЗАВЕРШЕНИЯ ТЕСТИРОВАНИЯ	9
7. ПЛАН-ГРАФИК ПРОВЕДЕНИЯ ТЕСТИРОВАНИЯ	10
ПРИЛОЖЕНИЕ А. ТЕСТ-КЕЙСЫ	12

ВВЕДЕНИЕ

1.1 Основная информация

Документ описывает методы и подходы к тестированию, которые будут использоваться тестировщиками отдела тестирования, для тестирования раздела «Лайки» приложения ВКонтакте. План тестирования может использоваться как тестировщиками, так и менеджерами, разработчиками.

Объект тестирования – это деятельность, направленная на проверку работоспособности функций сайта и приложения в браузерах Chrome, Safari и Firefox, операционных систем Android и iOS.

1.2 Цель

Цель написания тест-плана – описать подход к тестированию функциональности раздела лайков в мобильном приложении и веб-версии социальной сети ВКонтакте. Тестирование направлено на проверку корректности работы функции лайков, включая постановку, удаление, отображение количества лайков и уведомления. Для этого необходимо:

- убедиться, что пользователь может ставить и убирать лайки;
- проверить корректное обновление счетчика лайков;
- подтвердить, что уведомления о лайках отправляются и отображаются корректно;
- проверить работу функции при различных сценариях использования (офлайн, слабое соединение, большое количество лайков);
- убедиться в отсутствии багов в UI/UX;
- привести результаты тестирования.

Результаты будут отправлены в виде отчетов. Все найденные дефекты будут отслеживаться с помощью баг-трекинговой системы.

2. РАМКИ ПРОЕКТА

2.1 Объект тестирования

Тестирование модуля раздела лайков в приложении ВКонтакте.

Платформы: iOS, Android.

Типы тестирования: функциональное, UI/UX тестирование, нагрузочное и регрессионное тестирование.

Объекты тестирования:

- пользовательский интерфейс UI;
- тестирование backend и API;
- тестирование функционала раздела: постановка лайка на посты, комментарии, фото, видео, удаление лайка, отображение счетчика лайков, уведомления о лайках, взаимодействие с лайками в ленте, профиле и диалогах, фильтрация лайков по отображаемому контенту, хронологический порядок (от ранних к более поздним), проверка работы раздела приватности, производительность раздела при большом количестве лайков, синхронизация раздела на различных устройствах.

3. ПЛАН РАБОТЫ

В рамках проведения тестирования был принят следующий план работы:

1. Подготовка плана тестирования.
2. Согласование тест-плана.
3. Подготовка материалов необходимых для тестирования (документация, устройства).
4. Функциональное тестирование.
5. Проведение юзабилити-тестирования.
6. Тестирование Backend.
7. Проведение нагрузочного тестирования.
8. Формирование отчета о дефектах.
9. Передача документации разработчикам.
10. Подготовка финального отчета.

4. ПЛАН И СТРАТЕГИЯ ТЕСТИРОВАНИЯ

Процедура тестирования предполагает наличие сообщения об обнаруженных ошибках программного обеспечения. Необходимо проверить различные аспекты тестируемого программного обеспечения, для этого требуется выполнение различных типов тестирования.

Основные типы тестирования, которые будут выполнены:

- функциональное тестирование;
- тестирование пользовательского интерфейса;
- UI/UX -тестирование;
- регрессионное тестирование.

Также будет проверено, как сервис работает в браузерах и приложениях, в том числе desktop, которые должны его поддерживать, как он запускается и останавливается, сколько времени ему требуется для запуска.

В рамках тест-плана не будут выполняться виды тестирования:

- тестирование безопасности, так как данный функционал не взаимодействует с конфиденциальными данными пользователей, однако при необходимости могут быть проведены дополнительные тесты на защиту от накрутки лайков.

4.1 Функциональное тестирование

Цель: убедиться, что весь функционал раздела работает в соответствии с требованиями, и в разделе не появляется существенных дефектов. Функциональное тестирование является наиболее существенной частью тестирования программного обеспечения, включающее в себя проверку основных функций (постановка/снятие, лайков, фильтрация, пагинация).

Цель тестирования

Обеспечение надлежащего качества целевой функциональности

Техника	Выполните каждый сценарий, используя допустимые и недопустимые данные: <ul style="list-style-type: none">- ожидаемые результаты возникают при использовании достоверных данных- соответствующие сообщения об ошибках или предупреждения отображаются, когда используются неверные данные- каждый пункт проверен
Входные критерии	<ul style="list-style-type: none">- подготовлено тестовое окружение, приложение готово к тестированию на тестовой площадке
Критерий приемки	<ul style="list-style-type: none">- все дымовые тесты пройдены- нет блокирующих багов- все баги с высоким приоритетом поправлены- тесты пройдены

4.2 UI/UX тестирование

Цель: проверить пользовательские пути на предмет удобства использования. Определить, понимают ли пользователи как пользоваться сервисом «Лайки». Выявить сложности при работе с сервисом «Лайки». Узнать отношение пользователей к такому сервису. Понять на сколько удобно пользователю использовать сервис «Лайки».

Цель тестирования	Обеспечение надлежащего качества целевой функциональности
--------------------------	--

Техника	Выполните каждый сценарий, используя известные нам данные: - ожидаемые результаты возникают при использовании достоверных данных - соответствующие сообщения об ошибках или предупреждения отображаются, когда используются неверные данные - каждый пункт проверен
Входные критерии	- подготовлено тестовое окружение, функциональность готова к тестированию на тестовой площадке
Критерий приемки	- пользователь успешно прошел все тестовые сценарии - нет блокирующих багов - уведомления об ошибках работают корректно

4.3 Нагрузочное тестирование

Цель тестирования	Обеспечение надлежащего качества целевой функциональности
--------------------------	--

Техника	Выполните каждый сценарий, используя известные нам данные: - ожидаемые результаты возникают при использовании достоверных данных - соответствующие сообщения об ошибках или предупреждения отображаются, когда используются неверные данные - каждый пункт проверен
Входные критерии	- подготовлено тестовое окружение, функциональность готова к тестированию на тестовой площадке - доступны все необходимые сетевые и нагрузочные ресурсы
Критерий приемки	- непревышение времени работы функциональности - стабильная работа при больших нагрузках - быстрое восстановление при возникновении ошибок

4.4 Регрессионное тестирование

Цель тестирования	Обеспечение надлежащего качества целевой функциональности
--------------------------	--

Техника	Выполните каждый сценарий, используя известные нам данные: данные:
---------	---

Цель тестирования	Обеспечение надлежащего качества целевой функциональности
	<ul style="list-style-type: none"> - ожидаемые результаты возникают при использовании достоверных данных - соответствующие сообщения об ошибках или предупреждения отображаются, когда используются неверные данные - каждый пункт проверен
Входные критерии	<ul style="list-style-type: none"> - подготовлено тестовое окружение, функциональность готова к тестированию на тестовой площадке - отсутствуют изменения в коде или тестовых данных
Критерий приемки	<ul style="list-style-type: none"> - отсутствие ошибок по итогам тестирования - соответствие работы с заявленными требованиями

4.5 Сообщения об ошибках

Отчеты об ошибках должен быть информативен для команды разработчиков и давать возможность разработчику воспроизвести данный баг и понять его источник с дальнейшей возможностью его исправления.

Степень серьезности ошибок можно разделить на четыре категории:

Blocker (блокирующая) – блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы.

Critical (критическая) – это сбой всей программной системы или критической подсистемы, тестирование не может быть выполнено после возникновения дефекта. Это также относится к сбоям при потере данных и процессам, которые оставляют несогласованные данные, хранящиеся в базе данных.

Major (значительные) – ошибки, которые также приводят к выходу из строя всей или части системы, но существуют некоторые альтернативы обработки, которые позволяют продолжить работу системы. Это также относится к сбою или прерыванию работы системы во время нормальной работы не критического потока.

Minor (незначительные) – не приводят к сбою, но приводят к тому, что система показывает неправильные, неполные или противоречивые результаты.

Trivial (тривиальные) – это небольшие ошибки, которые не влияют на функциональность: опечатки, грамматические ошибки, неправильная терминология и т.д.

Информация, указанная в каждом отчете об ошибке:

- название программного продукта;
- платформа, на которой проводилось тестирование;
- номер версии программного продукта (при тестировании на мобильном устройстве);
- предусловие;
- описание с шагами воспроизведения;
- приложения (фото, видео, логи);
- указание приоритета и серьезности;
- браузер, в котором проводились тесты;
- разработчик.

5. ТРЕБОВАНИЯ К ОКРУЖЕНИЮ

Инструменты

Наименование процесса	Инструменты
Баг-трекинг-система	Jira
Тест-кейсы	TestRail
Анализ сетевых запросов	Charles Proxy/Fiddler
Тестирование бета-версий	TestFlight (iOS) / Google Play Beta (Android)
Эмуляторы	Xcode Simulator, Android Studio Emulator
Тестирование API	Postman
Захват изображений	ScreenShorts/Video capture

Устройства

Устройство	Операционная система
iPhone 14 pro	17.6.1
Samsung Galaxy S22	Android 13
Xiaomi Redmi Pad SE	Android 9

Список браузеров

Наименование браузера	Версия браузера
Chrome	v.136.0.7103.93 (arm64)
Safari	17.6.1

Firefox	v. 138.0.4
---------	------------

Версия приложения: последняя стабильная версия ВКонтакте.

6. КРИТЕРИИ ЗАВЕРШЕНИЯ ТЕСТИРОВАНИЯ

1) *Покрытие требований*

- все функциональные и нефункциональные требования, указанные в техническом задании, протестированы.
- выполнено 100% покрытие тест-кейсами для критических функций сервиса (постановка лайков, отображение счетчика лайков, удаление лайков).
- проверены все пользовательские сценарии, включая граничные случаи (например, максимальное количество лайков, ограничения по времени).

2) *Успешность выполнения тест-кейсов*

- успешно пройдено не менее 95% тест-кейсов для всех видов тестирования.
- все критические и блокирующие дефекты устранены.
- доля дефектов средней и низкой критичности не превышает 10% от общего числа тест-кейсов и не влияет на ключевую функциональность.

3) *Производительность и нагрузка*

- функциональность выдерживает нагрузку в более чем 1000 лайков в секунду без задержек.
- время отклика на действие пользователя (постановка/удаление лайка) не превышает 1 секунды при нормальной нагрузке.

4) *Совместимость*

- функциональность протестирована и корректно работает на всех платформах, на которых проводилось тестирование (см. раздел 5)
- подтверждена совместимость с последними версиями API

5) *Стабильность*

- функциональность работает стабильно в течение 48 часов непрерывного тестирования без сбоев и перезапусков.
- отсутствуют критические ошибки, приводящие к недоступности функциональности.

6) *Пользовательский опыт*

- подтверждена корректность отображения лайков и счетчиков на всех типах устройств и разрешений экрана
- положительные результаты юзабилити-тестирования.

7) Документация и обратная связь

- все обнаруженные дефекты заведены в баг-трекингтовую систему.
- получено подтверждение от заказчика/менеджера проекта о соответствии сервиса требованиям.
- подготовлен финальный отчет о тестировании, включающий результаты, статистику дефектов и рекомендации.

8) Временные рамки

- тестирование завершено в соответствии с утвержденным графиком тест-плана (см. раздел 7)
- все этапы тестирования выполнены согласно установленных сроков

7. ПЛАН-ГРАФИК ПРОВЕДЕНИЯ ТЕСТИРОВАНИЯ

Этап	Описание	Даты	Ответственный
Подготовительный	Анализ документации сервиса, определение функциональных и нефункциональных требований	02.06.2025 - 03.06.2025	Аналитик, QA-инженер
	Создание тест-кейсов для функционального, регрессионного и интеграционного тестирования	04.06.2025 - 06.06.2025	QA-инженер
	Настройка тестового стенда, подготовка тестовых аккаунтов ВКонтакте	09.06.2025 - 10.06.2025	DevOps, QA-инженер
Функциональное тестирование	Проверка корректности добавления лайков к постам, фото, видео	11.06.2025 - 11.06.2025	QA-инженер
	Проверка корректности удаления лайков, обновления счетчика	12.06.2025 - 12.06.2025	QA-инженер
	Проверка ограничений на количество лайков (дневные лимиты, антиспам)	13.06.2025	QA-инженер
UI/UX - тестирование	Оценка удобства интерфейса (доступность кнопок, интуитивность действий)	16.06.2025 - 17.06.2025	QA-инженер, UX-специалист
	Проведение тестирования с группой пользователей, анализ их отзывов	18.06.2025	UX-специалист
Тестирование производительности	Проверка работы сервиса при высоком количестве запросов (1000+)	19.06.2025 - 20.06.2025	QA-инженер, DevOps

Этап	Описание	Даты	Ответственный
	лайков/мин), измерение времени отклика		
	Определение предела нагрузки сервиса до отказа	23.06.2025	QA-инженер, DevOps
Тестирование обработки ошибок	Проверка поведения сервиса при некорректных данных (невалидные токены, превышение лимитов)	24.06.2025	QA-инженер
	Проверка отображения понятных сообщений об ошибках для пользователей	24.06.2025	QA-инженер
Финальный этап	Составление отчета о результатах тестирования, выявленных ошибках и их статусе	25.06.2025	QA-инженер
	Передача отчета заказчику/разработчикам, согласование релиза	27.06.2025	QA-инженер

ПРИЛОЖЕНИЕ А. ТЕСТ-КЕЙСЫ

Функциональное тестирование

ID	ТС-01
Сценарий	Постановка лайка на пост
Шаги воспроизведения	1. Открыть ленту новостей. 2. Выбрать пост. 3. Нажать на кнопку лайка.
Ожидаемый результат	Лайк отображается (иконка становится красной), счетчик лайков увеличивается на 1.
ID	ТС-02
Сценарий	Удаление лайка с поста
Шаги воспроизведения	1. Открыть пост с уже поставленным лайком. 2. Нажать на кнопку лайка.
Ожидаемый результат	Лайк снимается (иконка становится серой), счетчик лайков уменьшается на 1.
ID	ТС-03
Сценарий	Лайк на комментарий
Шаги воспроизведения	1. Открыть пост с комментариями. 2. Нажать на кнопку лайка у комментария.
Ожидаемый результат	Лайк отображается, счетчик лайков комментария увеличивается.
ID	ТС-04
Сценарий	Лайк на фото/видео
Шаги воспроизведения	1. Открыть фото/видео в альбоме. 2. Нажать на кнопку лайка.
Ожидаемый результат	Лайк отображается, счетчик обновляется.
ID	ТС-05
Сценарий	Уведомление о лайке
Шаги воспроизведения	1. Попросить другого пользователя поставить лайк на ваш пост. 2. Проверить раздел уведомлений.
Ожидаемый результат	Уведомление о лайке отображается с корректной информацией (имя пользователя, тип контента).

UI/UX-тестирование

ID	ТС-06
Сценарий	Отображение иконки лайка
Шаги воспроизведения	1. Открыть пост/фото/видео. 2. Проверить состояние иконки лайка (до и после нажатия).
Ожидаемый результат	Иконка отображается корректно, анимация нажатия плавная, цвета соответствуют дизайну.
ID	ТС-07
Сценарий	Отображение иконки лайка
Шаги воспроизведения	1. Открыть пост с лайками. 2. Поставить/убрать лайк.
Ожидаемый результат	Счетчик обновляется мгновенно, цифры отображаются корректно.
ID	ТС-08
Сценарий	Адаптивность на разных устройствах
Шаги воспроизведения	1. Открыть приложение на устройствах с разными разрешениями экрана. 2. Проверить отображение кнопки лайка и счетчика.
Ожидаемый результат	Элементы UI отображаются без смещений, текст читаем, кнопка кликабельна.

Нагрузочное тестирование

ID	ТС-09
Сценарий	Массовая постановка лайков
Шаги воспроизведения	1. Использовать скрипт/тестовый аккаунт для отправки 1000 лайков на один пост. 2. Проверить счетчик и производительность.
Ожидаемый результат	Счетчик лайков обновляется корректно, приложение не зависает.
ID	ТС-10
Сценарий	Лайки при слабом соединении
Шаги воспроизведения	1. Переключить сеть на 2G. 2. Поставить лайк на пост.

**Ожидаемый
результат**

Лайк сохраняется, отображается после восстановления соединения.

Тестирование ошибок

ID	ТС-11
Сценарий	Лайк в офлайн-режиме
Шаги воспроизведения	1. Отключить интернет. 2. Поставить лайк на пост.
Ожидаемый результат	Лайк отображается локально, синхронизируется после подключения к сети.

ID	ТС-12
Сценарий	Двойное нажатие на лайк
Шаги воспроизведения	1. Быстро дважды нажать на кнопку лайка.
Ожидаемый результат	Лайк ставится только один раз, счетчик не дублируется.

Requirements Traceability Matrix	Requierelement	Функциональное	UI/UX	Нагрузочное	Регрессионное
Test					
TC-01		X			
TC-02		X			
TC-03		X			
TC-04		X			
TC-05		X			
TC-06			X		
TC-07			X		
TC-08			X		
TC-09				X	
TC-10				X	
TC-11					X
TC-12					X