

Тест-план для мобильного приложения MarketHelp

1. Введение

Цель документа: Определить подход, объем и методы тестирования мобильного приложения MarketHelp для обеспечения соответствия требованиям, указанным в техническом задании.

Объект тестирования:

- Мобильное приложение MarketHelp (Android 10+);
- Интеграция с API Яндекс Маркета;
- ИИ-модель классификации отзывов и анализа тональности;
- Безопасность данных (API-ключи, пароли).

2. Типы тестирования

2.1 Функциональное тестирование:

- Авторизация/регистрация пользователей;
- Управление API-ключами (добавление, редактирование, удаление);
- Загрузка данных с Яндекс Маркета;
- Классификация отзывов по категориям (доставка, качество, упаковка);
- Анализ тональности (негатив/позитив);
- Генерация инфографики и экспорт в PNG;
- Работа фильтров (дата, рейтинг, категория);
- Роли пользователей (базовый/премиум тарифы, администратор).

2.2 Нефункциональное тестирование:

Производительность:

- Время загрузки данных ≤ 3 сек;
- Генерация инфографики ≤ 5 сек;
- Нагрузочное тестирование (20+ пользователей).

Безопасность:

- Шифрование паролей и API-ключей;
- Защита от SQL-инъекций.

Юзабилити:

- Интуитивность интерфейса (оценка $\geq 4/5$ по опросу пользователей).

Кросс-платформенность:

- Корректная работа на Android 10+.

2.3 Интеграционное тестирование:

- Взаимодействие фронтенда (Flutter) с бэкендом (Kotlin/Kora);
- Интеграция с ИИ-моделью (Qwen);
- Работа с API Яндекс Маркета.

3 Критерии входа/выхода

3.1 Критерии входа:

- Завершена разработка всех модулей;
- Проведено модульное тестирование;
- Готовы тестовые данные (API-ключи, отзывы, тестовые учетные записи).

3.2 Критерии выхода:

- Успешно пройдены все тестовые сценарии;
- Отсутствие критических дефектов (Severity S1);
- Удовлетворены нефункциональные требования (производительность, безопасность).

4 Тестовые сценарии

4.1 Авторизация/регистрация

- Ввод корректных логина/пароля → успешный вход;
- Ввод некорректных данных → отображение ошибки;
- Регистрация нового пользователя → создание учетной записи;

4.2 Управление API-ключами

- Добавление ключа → отображение в списке;
- Редактирование ключа → сохранение изменений;
- Удаление ключа → исключение из списка;
- Ввод невалидного ключа → ошибка валидации.

4.3 Генерация инфографики

- Применение фильтров (дата, рейтинг, категория) → корректное отображение данных;
- Экспорт в PNG → сохранение файла в галерее;
- Проверка водяных знаков для базового тарифа.

4.4 Работа ИИ-модели

- Классификация 50 тестовых отзывов → точность $\geq 85\%$;
- Анализ тональности → корректное определение негатива/позитива;

- Генерация рекомендаций → релевантность советов (например, "улучшите срок доставки").

4.5 Безопасность

- Проверка хеширования паролей в БД;
- Тестирование на SQL-инъекции → блокировка несанкционированных запросов.

5. Окружение и инструменты

5.1 Окружение:

- Android-устройства (версии 10+);
- Тестовый сервер бэкенда (Kotlin/Kora + PostgreSQL);
- Docker-контейнеры для развертывания.

5.2 Подходы и инструменты:

- Ручное тестирование: проверка UI, сценариев использования, юзабилити;
- Автоматизация: использование Appium для тестирования на Android, Postman для проверки API;
- Нагрузочное тестирование: JMeter для эмуляции 20 пользователей;
- Безопасность: OWASP ZAP для поиска уязвимостей;
- Локализация: проверка переводов через Crowdin.

6 Риски и минимизация

6.1 Риски:

- Низкая производительность ИИ;
- Утечка API-ключей;
- Ошибки интеграции с Яндекс Маркет;
- Падение приложения под нагрузкой.

6.2 Минимизация:

- Регулярное обновление модели Qwen актуальными данными;
- Шифрование ключей в БД, использование HTTPS;
- Настройка mock-сервера для тестирования API на этапе разработки;
- Горизонтальное масштабирование серверов через Docker.

7 Тестовые сценарии

7.1 Авторизация и регистрация

7.1.1 Успешная авторизация

Цель: проверить вход в систему с корректными данными.

Предусловие: пользователь зарегистрирован.

Шаги:

1. Открыть экран авторизации;
2. Ввести валидный логин и пароль;
3. Нажать кнопку «Войти».

Ожидаемый результат: открывается главный экран.

7.1.2 Неуспешная авторизация

Цель: проверить обработку ошибки при вводе неверного пароля.

Предусловие: пользователь зарегистрирован.

Шаги:

1. Ввести валидный логин и неверный пароль;
2. Нажать «Войти».

Ожидаемый результат: появляется сообщение «Неверный логин или пароль».

7.2 Регистрация нового пользователя

Цель: проверить создание аккаунта.

Предусловие: пользователь не зарегистрирован.

Шаги:

1. Перейти на вкладку «Регистрация»;
2. Заполнить поля: логин, пароль, подтверждение пароля;
3. Нажать «Зарегистрироваться».

Ожидаемый результат:

- Аккаунт создан, данные сохранены в БД;
- Пользователь перенаправлен на главный экран.

7.3 Управление API-ключами

7.3.1 Добавление нового API-ключа

Цель: проверить сохранение ключа и его отображение в списке.

Предусловие: пользователь авторизован.

1. Перейти в раздел «Новый магазин»;
2. Нажать «Добавить ключ»;
3. Ввести валидный ключ (например, YNDX-1234-ABCD) и название магазина;
4. Добавить.

Ожидаемый результат: ключ отображается в списке активных, т.е в разделе «Магазины» будут отображаться API-ключи=магазины.

7.3.2 Редактирование API-ключа

Цель: Проверить обновление данных ключа.

Предусловие: в системе есть активный ключ.

Шаги:

1. Выбрать ключ=магазин из списка;
2. Нажать на символ карандаша;
3. Ввести новый ключ;
4. Сохранить изменения.

Ожидаемый результат: старый ключ заменен новым.

7.3.3 Удаление API-ключа

Цель: проверить удаление ключа из системы.

Предусловие: в системе есть активный ключ=магазин.

Шаги:

1. Выбрать ключ=магазин из списка;
2. Нажать на символ карандаша;
3. Нажать «Удалить»;
4. Подтвердить действие. **У НАС ЕСТЬ ЭКРАН ПОДТВЕРЖДЕНИЕ?**

Ожидаемый результат: ключ=магазин исчезает из списка.

7.3.4 Ввод некорректного ключа

Цель: проверить валидацию формата ключа.

Шаги:

1. Ввести ключ в неверном формате (например, 1234ABCD);
2. Попытаться сохранить.

Ожидаемый результат: появляется ошибка: «Неверный формат ключа».
ЕСТЬ ТАКОЙ ЭКРАН ОШИБКИ?

7.4 Генерация инфографики