**Стартап під назвою “Мурчик”**

**Test Strategy**

**Revision History**

| Date | Version | Author | Description |
| --- | --- | --- | --- |
| 22.06.2025 | 1.0 | Команда QA “ Мурчиків” | Перша версія тестової стратегії |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**1. Scope 3**](#_heading=h.9h3ld38so808)

[**2. Test Approach 4**](#_heading=h.7y7i1y1aiyw8)

[**3. Test Environment 5**](#_heading=h.aoe6ji7ngl6j)

[**4. Testing Tools 6**](#_heading=h.kvxttt3zf5r3)

[**5. Release Control 7**](#_heading=h.9k1vosb9d2vu)

[**6. Risk Analysis 8**](#_heading=h.olhxb6qt1foa)

[**7. Review and Approvals 9**](#_heading=h.hioxlxpqjdc1)

# Scope

* Тестування функціональності мобільного застосунку "Мурчик"
* Перевірка стабільності та безпеки обміну фото з використанням позитивного та негативного тестування
* Дії: реєстрація, завантаження фото, коментування, вподобання, перегляд профілів
* Тестування проводиться по принципу Agile з використанням Scrum-майстра на кожному етапі розробки ПЗ.

Документ буде:

* 1. Переглянутий: **QA Lead, Product Owner**
  2. Затверджений: **Project Manager**

Тестування виконуватиметься за такими етапами спринтів:

* Раннє тестування
* Після завершення основних функцій (smoke)
* Перед релізом (full regression)
* При кожному фіксі (targeted testing)

# Test Approach

* **Процес:**
* Планування → Моніторинг та контроль→ Аналіз та дизайн → Написання тест-кейсів —-Виконання → Завершення тестування
* **Рівні тестування які ми залучаємо:**
  + Юніт (на боці девелоперів)
  + Інтеграційне ( проводиться розробниками в коді)
  + Системне (вся логіка роботи додатку)
  + Приймальне тестування ( Beta- testing)
* **Типи тестування:**
  + Функціональне, UI, Регресійне, Кросплатформене, Нефункціональне (load testing, usability testing, security testing)
* **Інструменти:**
  + Тест-кейси – TestRail, Google Sheets
  + Репорти – Jira
  + Часткова автоматизація для smok тестування
* **Ролі:**
  + QA Engineer — виконання ручного/автоматизованого тестування
  + QA Lead — планування, контроль якості, затвердження
  + Dev Team — виправлення дефектів
  + Product Owner — прийняття функцій після тестування
* **Дефекти:**
  + Створення — через Jira
  + Тріаж — щодня, спільно з розробниками
  + Ретест і регресія — згідно з графіком

# Test Environment

У нас кілька середовищ для різних цілей:

* **Dev-середовище:** для перевірки нових функцій у режимі розробки. Необхідне: доступ до внутрішньої бази даних, журналів помилок, інструментів девелопера.
* **QA-середовище:** для ручного та автоматизованого тестування. Має стабільну копію бази, доступ до серверів авторизації, зберігання зображень та push-нотифікацій.
* **Prod-середовище:** робоче середовище для smoke-тестів перед виходом. Має підключення до фінальних API, бази та реального сховища фото.

**Кількість налаштувань/вимог:**

* Кожне середовище потребує:
  + Активної бази даних (1 шт)
  + Серверів API (2 інстанси)
  + Сховища зображень (1 bucket у S3 або Firebase)
  + Конфігурацій для пуш-сповіщень

**Стратегія резервного копіювання:**

* QA-середовище:
  + Щотижневий автоматичний бекап тестової бази
  + Окремий бекап перед великим релізом або міграцією
  + Відновлення — через скрипт SQL + відновлення файлів з S3 архіву
* Dev-середовище:
  + Бекапи лише вручну перед критичними змінами
* Prod-середовище:
  + Зовнішня стратегія бекапу під контролем DevOps (не покривається QA)

# 

**Інструменти, які ми використовуємо 🛠️**

* **TestRail / Google Sheets** — плануємо й ведемо тести
* **Jira** — керуємо задачами й багами
* **Appium** — автоматизуємо перевірки на Android/iOS
* **Firebase** — моніторимо краші

**Інструменти з відкритим кодом:**

* **Appium (4 користувачі)** — автоматизація мобільного тестування
* **Allure Reports (4 користувачі)** — формування автоматизованих звітів
* **Postman (використовується безкоштовна версія, 5 користувачів)** — тестування API

**Комерційні інструменти:**

* **TestRail (3 ліцензії)** — керування тест-кейсами
* **Jira (5 ліцензій)** — трекінг завдань і багів

**Висновки з планування:**

* Для зростаючої команди QA потрібне масштабування TestRail (до 5 користувачів).

# Release Control

### Як ми контролюємо релізи

* Кожен реліз має номер (наприклад: m.2.0 )
* Перед релізом:
  + smoke-тест (чи все живе?)
  + регресія (чи все залишилось на місці?)
* QA Lead та Product Owner підписують фінальне «можна»

# Risk Analysis

**Ризики:**

* Зникли тестові дані
* щотижневі бекапи, контроль доступів
* Помилки нових фічах на проді
* повне регресійне тестування перед релізом
* Недостатнє покриття тестами
* Тест дизайн за user-stories

# Review and Approvals

* All these activities are reviewed and sign off by the business team, project management, development team, etc.
* Summary of review changes should be traced at the beginning of the document along with approved date, name, and comment