# Cahier des charges – Application Android Simplifiée (Permis de pêche)

**Version** : 1.0  
**Date** : 23 août 2025  
**Portée** : Application Android (agents) + Back‑office web (administration)

**1) Objectif & périmètre**

Mettre en place une solution **rapide, fiable et simple** pour :

* **Collecter** les informations des pêcheurs sur le terrain via **une application Android** unique avec **un formulaire fixe** (Permis de pêche) ;
* **Générer automatiquement** une **carte de permis** (publipostage sur un modèle fixe) avec **photo** et **QR Code sécurisé** ;
* **Vérifier l’authenticité** d’une carte **uniquement via l’application officielle** (scan intégré) ;
* **Gérer, modifier, visualiser et exporter** les données et cartes via un **back‑office administrateur** ;
* **Fonctionner hors ligne** côté agents, avec **synchronisation** vers la base centrale dès que le réseau est disponible.

**Hors périmètre (version simplifiée)** : multi-formulaires, GPS, workflows d’approbation complexes, comptes multiples côté agents (si non exigé), portail public.

**2) Rôles & responsabilités**

**2.1 Agent (Application Android)**

* Saisie des dossiers pêcheurs (formulaire unique) ;
* Prise de **photo d’identité** ;
* **Aperçu** immédiat de la carte générée ;
* **Scan** et **contrôle** des cartes via QR Code sécurisé ;
* Fonctionnement **offline‑first** avec synchronisation ultérieure.

**2.2 Administrateur (Back‑office web)**

* Accès sécurisé (identifiant/mot de passe) ;
* Consultation, recherche, **modification** des dossiers ;
* **Génération / prévisualisation** des cartes ;
* **Export** PDF **individuel (pdf en 2pages)** et **par lot** (plusieurs cartes dans un même document), export **PNG/JPG** ;
* Suivi de la synchronisation, sauvegardes et journal des opérations ;
* Gestion basique de la qualité : détection/traitement manuel des doublons.

**3) Formulaire unique « Permis de pêche »**

**3.1 Champs (exigés)**

* Identité : **Nom**, **Prénoms**, **Date de naissance** ;
* Coordonnées : **Adresse complète**, **Quartier/Village**, **Numéro de téléphone** ;
* Permis : **Type de pêche**, **Zone de pêche**, **Numéro XXX/MAEP**, **Numéro de série**, **Date de délivrance**, **Date d’expiration** ;
* Références : **Disponibilité IFU** (oui/non), **Numéro IFU** ou **Numéro RAVIP** si IFU indisponible ;
* Embarcation : **Nom d’embarcation**, **Numéro d’embarcation**, **Site de débarquement**, **Site d’habitation** ;
* Technique : **Type d’engin**, **Technique de pêche**, **Espèces ciblées**, **Permis de catégorie…** ;
* **Photo d’identité** ;
* **Horodateur** (généré automatiquement) et **Fait à**.

**3.2 Règles de validation**

* Formats : **date** (JJ/MM/AAAA), **téléphone** (règles pays), **numéros** (masques ex. XXX/MAEP-######) ;
* Champs obligatoires : identité, téléphone, numéros clés, dates, photo ;
* Unicité (côté base) : **Numéro de série** par défaut unique ;
* Gestion des noms/prénoms longs : **troncature intelligente** ou **réduction de police** sur la carte.

**4) Carte & publipostage**

**4.1 Modèle de carte**

* **Design fixe** (image ou HTML/CSS) fourni ;
* Zones définies : textes (placements, tailles min/max), **photo**, **QR Code** ;
* Rendu **aperçu** sur mobile et back‑office ;
* Export **PDF individuel**, **PDF par lot** (grille **2×5** = **10 cartes/page**), **PNG/JPG** ;
* Paramètres : DPI (300 conseillé), marges, traits de coupe, fond perdu.

**4.2 Moteur de publipostage**

* Mapping champ → zone ;
* Règles de **réduction de police** et **multi‑lignes** ;
* Recadrage photo (ratio configurable) ;
* Génération rapide (< 1 s par carte en aperçu sur matériel récent).

**5) QR Code sécurisé & contrôle**

**5.1 Objectif**

Garantir que **seule l’application officielle** puisse consulter les informations détaillées liées au QR Code ; empêcher l’accès aux données via un scan tiers ou un navigateur public.

**5.2 Contenu du QR Code (recommandé)**

* payload minimal **non sensible** :
  + **id** : identifiant dossier (UUID) ;
  + **sig** : **signature numérique** (ECDSA/Ed25519) calculée côté serveur sur {id|timestamp|version} ;
  + **ts** : horodatage d’émission ;
  + **v** : version du schéma QR.
* Encodage : JSON compacté puis Base45/URL‑safe.

**5.3 Lecture & vérification**

* **Application Android** embarque la **clé publique** de vérification ;
* Au scan :
  1. Décodage du QR ;
  2. **Vérification cryptographique** de sig ;
  3. Si signature **valide** :
     + **Online** : requête API pour récupérer l’état actuel du dossier ;
     + **Offline** : recherche dans **cache local** synchronisé (infos essentielles, statut, dates).
  4. Si signature invalide : carte **suspecte** (alerte fraude).

**5.4 Deep‑link privé (option supplémentaire)**

* Utiliser un **schéma privé** peche://verify?d=... **non reconnu par les navigateurs** ;
* Empêche l’ouverture dans une app générique ; seul notre scanner interne sait interpréter.

**5.5 Accès restreint**

* L’accès aux informations détaillées via scan est **réservé aux utilisateurs authentifiés** (agents/administrateurs) ;
* Journalisation : date, agent, résultat (valide/expiré/suspect).

**6) Application Android (Agent)**

**6.1 Fonctionnalités**

* **Créer / Modifier / Supprimer** un dossier ;
* **Prendre photo** avec guidage (cadrage visage, vérification netteté) ;
* **Aperçu** de la carte (moteur embarqué) ;
* **Scanner** un QR pour contrôles (validité, correspondance identité/numéro, date d’expiration) ;
* **Offline‑first** : stockage local (Room/SQLite), file de synchronisation ;
* **Recherche** : nom, téléphone, numéro, ID.

**6.2 Expérience & ergonomie**

* Formulaire **en étapes** ;
* Avertissements clairs en cas d’erreur de format ;
* Possibilité de reprendre un **brouillon** ;
* Temps d’aperçu < 1 s ; prise de photo fluide.

**6.3 Synchronisation**

* **Automatique** dès réseau disponible + bouton **forcer** la sync ;
* Conflits : stratégie **last‑write‑wins** ;
* Journal local des opérations.

**7) Back‑office Web (Administrateur)**

**7.1 Accès & sécurité**

* HTTPS, identifiant/mot de passe (hashés) ;
* Rôle : **Administrateur** ;
* Journalisation : connexions, exports, modifications.

**7.2 Gestion des dossiers**

* Liste, filtres (par zone, date, statut), recherche ;
* **Modification** des enregistrements ;
* **Historique** des changements (qui/quand/quoi) ;
* **Détection simple de doublons** (nom + téléphone + numéro de série).

**7.3 Cartes & exports**

* **Prévisualisation** de la carte (publipostage) ;
* **Export PDF individuel** ;
* **Export PDF par lot** (sélection, filtre, ou plusieurs cartes dans un même document) ;
* **Export image** (PNG/JPG) ;
* **Export données** (CSV/Excel) ;
* Paramètres d’export : DPI, marges, traits de coupe.

**7.4 Synchronisation & sauvegardes**

* État des files de synchronisation (succès/échecs) ;
* **Export/backup** de la base (CSV/JSON, dumps planifiés) ;
* Restauration depuis sauvegarde (procédure documentée).

**8) Architecture & technologies (proposées)**

* **Mobile (Android)** : Kotlin + **Jetpack Compose**, **CameraX**, **ZXing** (scan QR), **Room** (SQLite), **WorkManager** (sync) ;
* **Back‑end** : **Node.js (NestJS)** + **PostgreSQL**, **MinIO/S3** pour médias, **Puppeteer** pour PDF ;
* **Signature** : **Ed25519** (libsodium/tweetnacl) ; clé **privée** côté serveur, **publique** dans l’app ;
* **Back‑office** : **React/Next.js** + Tailwind (UI épurée) ;
* **Déploiement** : Docker (API, Web, DB, MinIO), serveur local ou cloud (VPS).

**9) Modèle de données (simplifié)**

* **Fisher**(id, nom, prenoms, date\_naissance, telephone, adresse, quartier\_village)
* **Permit**(id, fisher\_id, type\_peche, zone\_peche, num\_maep, num\_serie [unique], date\_delivrance, date\_expiration, categorie, ifu\_disponible, num\_ifu, num\_ravip)
* **Vessel**(id, fisher\_id, nom\_embarcation, num\_embarcation, site\_debarquement, site\_habitation)
* **Media**(id, permit\_id, type = photo\_identite, url, meta)
* **Card**(id, permit\_id, qr\_payload, qr\_signature, horodateur, fait\_a, version)
* **User**(id, email, password\_hash, role = admin)
* **ScanLog**(id, card\_id, agent\_id/null, date, resultat, mode = online/offline)

**10) API (exemples minimaux)**

* POST /api/permits : créer dossier + médias ;
* GET /api/permits/{id} : lire ; PUT /api/permits/{id} : modifier ;
* POST /api/cards/{permitId}/generate : produire carte + QR ;
* GET /api/cards/{id}/pdf : obtenir PDF individuel ;
* POST /api/cards/export : PDF par lot (filtres) ;
* POST /api/scans/verify : vérification QR (online) ;
* GET /api/export/csv : export données.

**11) Sécurité & conformité**

* **Confidentialité** : aucune PII en clair dans le QR ;
* **Intégrité** : signature Ed25519 vérifiée par l’app ;
* **Authentification** : requise pour lecture détaillée via app et back‑office ;
* **Transport** : HTTPS ;
* **Stockage** : mots de passe hashés (Argon2/BCrypt), photos sur stockage objet ;
* **Journalisation** : connexions, exports, scans ;
* **Sauvegardes** : dumps chiffrés, rétention 30–90 jours en local ; sauvegarde sur serveur en ligne illimitée
* **RGPD‑like** : accès/rectification/suppression sur demande (procédure).

**12) Performances & volumétrie**

* Cible : **plusieurs centaines de cartes**;
* Scan & vérification : < 500 ms (offline) ; < 1 s (online, réseau normal) ;
* Génération PDF lot 100 cartes : < 30 s sur serveur milieu de gamme (indicatif).

**13) Critères d’acceptation (MVP)**

1. Création, modification et recherche d’un dossier depuis l’app ;
2. Prise de photo, **aperçu** carte ;
3. Génération d’une carte avec **QR Code signé** ;
4. Scan dans l’app :
   * Rejet si signature invalide ;
   * Affichage des infos si valide (offline si en cache, online sinon) ;
5. Back‑office :
   * Consultation, **édition** ;
   * **Prévisualisation** et **exports** (PDF individuel, PDF par lot 10/page, PNG/JPG) ;
6. Aucune donnée personnelle accessible en dehors de l’app via le QR.

**14) Tests & recette**

* **Unitaires** : validations champs, masques, signature/verification QR ;
* **Intégration** : génération PDF/PNG, stockage médias, synchro ;
* **E2E** : parcours complet (saisie → carte → scan → export) ;
* **Offline** : création et scan sans réseau ;
* **Sécurité** : tentative d’ouverture QR via navigateur/app tierce (doit échouer).

**15) Livrables**

* APK Android + code source ;
* Code source Back‑end + Back‑office (Docker) ;
* Template de carte (image/HTML) + mapping ;
* Documentation d’installation & d’utilisation ;
* Procédure de sauvegarde/restauration ;
* Jeu d’essai (10–20 dossiers) ;
* Plan de tests & PV de recette.

**Liste des CHAMPS nécessaires (Liste à valider) :**

* Horodateur
* Fait à
* Type de Pêche
* Zone de Pêche
* Numéro XXX/MAEP
* Numéro de série
* Date d'Expiration
* Nom
* Prénoms
* Numéro de téléphone
* Disponibilité de l'IFU
* Numéro IFU
* Numéro RAVIP
* Adresse complète
* Quartier/Village
* Numéro de téléphone
* Nom d'embarcation
* Numéro d'embarcation
* Site de débarquement
* Site d'habitation
* Type d'engin
* Technique de pêche
* Espèces Ciblées
* Permis de catégorie...
* Photo d'identité