**Rapport - Application de Surveillance Écologique (Halfaoui Mohand said – Khaber said).**

**Description Générale**

L'application développée permet la surveillance d'espèces protégées dans des parcs naturels via une interface Android intuitive. Elle collecte des données environnementales en temps réel et génère des alertes lorsque les seuils écologiques sont dépassés.

**Fonctionnalités principales implémentées :**

**Gestion des espèces**

* Création, modification et suppression d'espèces avec informations détaillées :
  + Nom de l'espèce
  + Statut de conservation
  + Population estimée
  + Adresse et coordonnées GPS
  + Seuils minimaux/maximaux de température et d'humidité

**Capteurs environnementaux**

* Lecture de la température et de l'humidité depuis les capteurs du téléphone (si disponibles)
* Boutons pour copier rapidement la valeur courante vers les seuils min/max
* Panneau d'aperçu temps réel des mesures avec fallback si capteurs absents

**Géolocalisation**

* Récupération de la position GPS actuelle (latitude, longitude) via getBestLocation()
* Conversion en adresse lisible via géocodage inversé
* Bouton "My Location" pour remplir automatiquement l'adresse et les coordonnées
* Cache intelligent pour optimiser les performances

**Alertes automatiques**

* Vérification en temps réel si la température ou l'humidité dépasse les seuils définis
* Notification locale immédiate en cas de dépassement lors de l'enregistrement
* Messages d'alerte avec conseils d'action corrective

**Tri et recherche avancés**

* Recherche d'espèces par nom avec debounce pour fluidité
* Tri par :
  + Nom (A–Z ou Z–A)
  + Plus récent / Plus ancien
  + Température maximale (croissant/décroissant)
  + Population

**Sauvegarde et synchronisation cloud**

* Backup : envoi de toutes les données locales vers Firestore
* Restore : récupération depuis Firestore et remplacement des données locales
* Synchronisation bidirectionnelle (syncUp/syncDownReplace)

**Interface utilisateur responsive**

* Interface moderne avec Jetpack Compose
* Écran d'accueil personnalisé avec message selon l'heure ("Bonjour/Bonsoir")
* Grille automatique sur grands écrans (Pixel C/paysage), liste sur mobiles
* Validation complète des formulaires avec messages d'erreur explicites
* Utilisation d'OutlinedTextField, ElevatedCard pour une UI claire

**Structures de Données**

**Modèle Species/SpeciesEntity**

data class Species(

val name: String,

val status: String,

val population: Int,

val minTemp: Double, val maxTemp: Double,

val minHumidity: Double?, val maxHumidity: Double?,

val latitude: Double, val longitude: Double,

val address: String,

val createdAt: Long

)

**Justification :** Cette structure englobe toutes les informations essentielles pour la surveillance écologique. Les seuils min/max permettent la validation des conditions environnementales, tandis que les coordonnées GPS facilitent la géolocalisation des habitats.

**Base de Données (Room)**

* **SpeciesDao** : opérations CRUD étendues (searchByName, getById, deleteById)
* **Repository** : couche d'abstraction gérant la synchronisation locale/cloud
* **Fonctions clés** : checkThresholds() pour la validation des seuils, syncUp/syncDownReplace pour Firebase

**Architecture Technique**

**Technologies utilisées :**

* **Langage** : Kotlin
* **UI** : Jetpack Compose (moderne et déclarative)
* **Base locale** : Room (SQLite) pour persistance hors ligne
* **Cloud** : Firebase Firestore pour synchronisation
* **Capteurs** : Android SensorManager (température/humidité)
* **Géolocalisation** : FusedLocationProviderClient
* **Notifications** : NotificationManagerCompat

**Pattern Repository :** Couche d'abstraction gérant la synchronisation entre Room (local) et Firestore (cloud), avec mapping automatique entre modèles domaine et entités de base de données.

**Discussion**

**Points Forts Réalisés**

* **Architecture robuste** : séparation claire des responsabilités avec pattern Repository
* **UX optimisée** : interface responsive, validation temps réel, recherche fluide avec debounce
* **Fiabilité** : persistance locale + synchronisation cloud, gestion gracieuse des capteurs manquants
* **Performance** : cache intelligent pour géolocalisation, interface adaptive selon taille d'écran

**Limitations Techniques Rencontrées**

**Fonctionnement Global**

1. **Saisie des données** : L'utilisateur saisit ou modifie les informations d'une espèce via le formulaire avec validation en temps réel (cohérence min≤max, humidité 0-100%)
2. **Collecte automatique** : Les capteurs (si disponibles) fournissent température et humidité actuelles, copiables vers les seuils via boutons dédiés
3. **Géolocalisation intelligente** : Récupération de l'adresse et coordonnées GPS via le bouton "My Location" avec cache pour performances optimales
4. **Sauvegarde et contrôle** : Lors de l'enregistrement, les données sont stockées dans Room et les seuils vérifiés automatiquement
5. **Alertes temps réel** : Si les mesures actuelles dépassent les seuils définis, notification immédiate avec conseils d'action
6. **Synchronisation cloud** : Backup vers Firestore et restore depuis le cloud pour sécuriser les données
7. **Consultation optimisée** : Liste responsive avec recherche par nom et tri multicritères pour navigation fluide

**Limitations Rencontrées**

* **Multilinguisme** : configuration présente mais non entièrement fonctionnelle
* **Capteur d'humidité** : indisponible sur l'émulateur SDK (limitation technique, non erreur de code)

**Améliorations Futures**

* Finalisation du système de langues (français/anglais)
* Carte interactive avec marqueurs d'habitats
* Notifications géolocalisées avancées
* Mode hors ligne avec synchronisation différée

**Conclusion**

L'application répond aux objectifs principaux du projet en offrant une solution complète de surveillance écologique. Malgré quelques limitations techniques liées à l'environnement de développement, l'ensemble des fonctionnalités critiques est opérationnel et l'architecture permet une évolutivité future.