

BTS SIO – Option SLAM

Solutions Logicielles et Applications Métiers
Session 2026

PROJET PERSONNEL ENCADRÉ

Dossier de réalisation professionnelle

GSB Appli-CR Mobile

Application mobile Android d'enregistrement et de suivi des comptes-rendus de visite médicale

THOMAS Paul

N° candidat : 02542614838

N° de réalisation : 2

Application en ligne : paul-padovani-thomas.com

Dépôt GitHub :

github.com/Daifuku420/App_GSB_Mobile

Table des matières

1. Fiche descriptive de réalisation professionnelle

2. Contexte de l'application

- 2.1 Présentation de Galaxy Swiss Bourdin
- 2.2 Objectifs de l'application mobile
- 2.3 Public cible et utilisateurs

3. Cahier des charges

- 3.1 Fonctionnalités attendues
- 3.2 Contraintes techniques
- 3.3 Budget et planning prévisionnel

4. Conception des données

- 4.1 Architecture client-serveur et API REST
- 4.2 Structure du projet Android
- 4.3 Base de données partagée (MCD)

5. Gestion de projet

- 5.1 Méthodologie et outils
- 5.2 Suivi d'avancement avec Jira

6. Mode opératoire global de l'application

- 6.1 Rôle Visiteur
- 6.2 Rôle Délégué régional

7. Documentation technique

- 7.1 Environnement et technologies
- 7.2 Communication avec l'API
- 7.3 Gestion de l'authentification
- 7.4 Sécurité

8. Tests et validation

9. Bilan et évolutions futures

10. Accès à l'application

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 1
Nom, prénom : THOMAS Paul		N° candidat : 02542614838
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>	Date : 15/06/2026
Organisation support de la réalisation professionnelle La réalisation s'inscrit dans le cadre d'un Projet Personnalisé Encadré, fondé sur un contexte fourni par le Centre de formation. Il concerne une entreprise pharmaceutique fictive, nommée Laboratoire Galaxy Swiss Bourdin (GSB). L'objectif était de concevoir des outils destinés à améliorer le suivi de l'activité de visite Médicale.		
Intitulé de la réalisation professionnelle Développement PHP de l'application de gestion de comptes-rendus de visiteurs médicaux		
Période de réalisation : sept. 2025 – Février 2026 Lieu : Eductive-Lyon (Sciences-u Lyon) Modalité : Seul		
Compétences travaillées <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données		
Conditions de réalisation⁵ (ressources fournies, résultats attendus) Ressources fournies : Contexte GSB de l'éducation nationale, Cahier des charges dont celle de l'éducation nationale, code existant, normes de développement, description du contexte, expression des besoins, script de la BDD existante, des ressources de formation, le liens utiles... Résultats attendus : Réalisation d'une application de gestion de fiches avec la réécriture du code front office, évolution de la BDD, création du code back office, accès avec authentification, doc technique, doc utilisateur, tests et contrôles...		
Description des ressources documentaires, matérielles et logicielles utilisées⁶ <ul style="list-style-type: none"> • Serveur local : XAMPP pour le développement local. • Éditeur de code : Visual Studio Code, intégré avec PHPMyAdmin pour la gestion de bases de données MySQL. • Langages de programmation : <ul style="list-style-type: none"> • Front-end: Java, xml. • Back-end: PHP, postgresql. • Bibliothèques : Bootstrap pour le design • Modélisation de données : Draw.io pour la conception du MCD • Gestion de projet : Teams pour les discussions, GitHub, tableau excel pour gestion • Supports supplémentaires : Une page de ressources avec documentation, formations, exemples, et liens utiles fournie par le professeur encadrant 		
Modalités d'accès aux productions⁷ et à leur documentation⁸ Accès au projet GSB : paul-padovani-thomas.com		

⁵ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

⁶ Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

⁷ Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve.* ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

**ANNEXE VII-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)****Épreuve E6 - Conception et développement d'applications (option SLAM)****Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs**

Notre projet consistait à développer une application web intuitive et performante destinée à moderniser la gestion des comptes-rendus de visite des visiteurs médicaux du laboratoire GSB. L'objectif était de remplacer l'ancienne application Access par une solution Web structurée autour d'une interface front-end ergonomique et d'une logique back-end robuste permettant la centralisation, la sécurisation et l'analyse des données de visite.

L'application devait permettre aux utilisateurs de saisir des comptes-rendus de visite, de consulter un historique sur plusieurs années, d'accéder aux données des praticiens et aux informations sur les produits, et de visualiser des statistiques liées à leur activité. Le système devait également intégrer un mécanisme d'authentification distinguant plusieurs niveaux de droits, notamment le Visiteur et les niveaux hiérarchiques, afin de sécuriser l'accès aux données et d'adapter les fonctionnalités proposées.

Pour la conception de l'interface utilisateur, j'ai utilisé un template Bootstrap, dont la flexibilité et les composants réactifs m'ont permis de créer une interface claire et conforme aux attentes du projet. Côté serveur, j'ai développé le back-end en PHP afin de gérer les requêtes et les traitements, tandis que l'accès aux données était assuré par des requêtes SQL reposant sur une base de données que j'ai adaptée à partir du modèle Access fourni. J'ai mis en place une authentification sécurisée, la gestion des sessions, le contrôle d'accès aux différentes pages selon le rôle de l'utilisateur, ainsi que les formulaires complets permettant la saisie des comptes-rendus. J'ai également actualisé la structure de la base de données pour intégrer toutes les contraintes du cahier des charges.

J'ai réalisé une application web fonctionnelle intégrant une interface responsive basée sur Bootstrap, un système de connexion sécurisé, un module complet de saisie des comptes-rendus de visite, un historique consultable et des pages dédiées aux statistiques. J'ai produit le schéma final de la base de données, les scripts SQL, l'architecture PHP et la gestion des droits utilisateurs. Enfin, j'ai rédigé une documentation technique, un guide utilisateur et j'ai mené les tests nécessaires pour garantir le bon fonctionnement de l'application.

- **Administrateur** : J'ai développé un espace administrateur permettant de consulter les logs d'activité pour assurer un suivi précis des actions réalisées, de créer de nouveaux comptes en attribuant les rôles appropriés, de modifier les visiteur/médicaments/praticiens existants.
- **Page d'accueil** : J'ai conçu une page d'accueil affichant les informations essentielles des utilisateurs (nom, prénom et matricule) ainsi qu'un tableau de bord pour avoir un accès facile et des mini logs (comme compte rendu récent et alerte quand périodicité dépassé 8 mois et que le visiteur doit retourner voir le praticien)
- **Gestion des utilisateurs** : J'ai mis en place un système de gestion des utilisateurs dans lequel le responsable de secteur peuvent absolument tout modifier, les délégués régionaux /visiteurs peuvent modifier leur propres informations.

⁸ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

2. Contexte de l'application

2.1 Présentation de Galaxy Swiss Bourdin

Galaxy Swiss Bourdin (GSB) est un laboratoire pharmaceutique international issu de la fusion entre le géant américain Galaxy, spécialisé dans les traitements contre les maladies virales, et le conglomérat européen Swiss Bourdin, expert en médicaments d'usage courant. Le siège administratif de l'entité européenne est installé à Paris, tandis que le siège social mondial se trouve à Philadelphie.

L'entreprise s'appuie sur une force commerciale de 480 visiteurs médicaux en France métropolitaine et 60 en outre-mer, organisés en une structure hiérarchique régionale à trois niveaux : visiteurs, délégués régionaux et responsables de secteur. Le contexte détaillé (historique, organisation, enquête utilisateurs de 2012) est décrit dans le dossier de la réalisation n°1 (GSB Appli-CR Web).

2.2 Objectifs de l'application mobile

L'application web GSB Appli-CR a répondu aux besoins de centralisation des comptes-rendus. Cependant, les visiteurs médicaux passent l'essentiel de leur temps sur le terrain. Ils n'ont pas toujours accès à un ordinateur pour saisir leurs CR immédiatement après chaque visite, ce qui entraîne un risque de perte d'informations ou de saisie tardive.

L'application mobile vise à résoudre ce problème :

- Saisie immédiate des comptes-rendus depuis le terrain
- Accès rapide au tableau de bord et à l'historique
- Consultation des praticiens et médicaments avant une visite
- Synchronisation en temps réel avec l'application web via API REST
- Interface tactile optimisée pour un usage nomade
- Même niveau de sécurité et de gestion des droits

2.3 Public cible et utilisateurs

- Le Visiteur médical : saisit ses CR après chaque visite, consulte son historique et son tableau de bord
- Le Délégué régional : supervise l'activité de son équipe, filtre les CR par visiteur ou par période
- Le Responsable de secteur : administration globale (gestion des utilisateurs, praticiens, médicaments)

3. Cahier des charges

3.1 Fonctionnalités attendues

L'application mobile reprend l'ensemble des fonctionnalités de l'application web, adaptées à l'ergonomie Android.

Fonctionnalités du rôle Visiteur

- Authentification sécurisée par identifiant et mot de passe
- Tableau de bord personnel : CR du mois, visites totales, coefficient de notoriété moyen, date de dernière visite
- Saisie d'un nouveau compte-rendu : date, praticien (Spinner), motif, médicament présenté, case à cocher échantillon, bilan textuel
- Consultation de l'historique personnel avec détail de chaque CR
- Accès aux ressources : liste des praticiens et catalogue des médicaments
- Consultation du profil personnel (matricule, rôle, région)

Fonctionnalités du rôle Délégué régional

En plus des fonctionnalités visiteur, le délégué régional accède aux comptes-rendus de tous les visiteurs de sa région, avec affichage par date et par motif.

Fonctionnalités du rôle Responsable de secteur

Le responsable de secteur dispose du niveau d'accès le plus élevé : consultation globale de tous les CR, gestion des utilisateurs, des praticiens et des médicaments.

3.2 Contraintes techniques

Architecture

L'application communique avec le même back-end PHP/MySQL que l'application web, via des API REST. Aucune base de données locale n'est embarquée : toutes les données transitent par le réseau. L'application nécessite donc une connexion internet.

Compatibilité

SDK minimum 24 (Android 7.0 Nougat), SDK cible 34 (Android 14). Interface optimisée pour smartphones en orientation portrait.

Sécurité

Authentification obligatoire, vérification des droits selon le rôle, requêtes préparées PDO côté serveur, données transmises en POST, mots de passe hachés bcrypt.

3.3 Budget et planning prévisionnel

Le budget global de développement de l'application mobile a été estimé à environ 22 000 € TTC. Ce montant, supérieur à celui de l'application web (12 000 €), s'explique par plusieurs facteurs propres au développement mobile :

- **Développement d'un back-end API dédié (5 000 €)** : bien que l'application web existante dispose déjà d'une base de données, il a fallu concevoir et développer l'ensemble des API REST en PHP pour exposer les données au format JSON. Chaque fonctionnalité (authentification, CRUD

comptes-rendus, gestion des utilisateurs, praticiens, médicaments, échantillons, statistiques du tableau de bord) nécessite un ou plusieurs endpoints dédiés avec validation des données, gestion des erreurs et sécurisation.

- **Développement de l'application Android native (8 000 €)** : le développement mobile natif en Java/Android est plus coûteux que le développement web. Chaque écran est une Activity avec son layout XML, sa logique de communication Volley, son parsing JSON et sa gestion d'état. L'application compte une douzaine d'écrans distincts, chacun avec des composants Material Design spécifiques (CardView, RecyclerView, Spinner, DatePicker).

- **Maintenance future assurée sur 2 ans (3 000 €)** : ce poste couvre la maintenance corrective et évolutive de l'application sur une période de deux ans après la livraison. Il inclut la correction des anomalies signalées par les utilisateurs, les mises à jour de compatibilité avec les nouvelles versions d'Android (évolution du SDK cible), l'adaptation aux éventuelles modifications de l'API back-end, ainsi que les ajustements mineurs d'ergonomie ou de performances demandés par le commanditaire.

- **Gestion de la synchronisation et cohérence des données (2 500 €)** : assurer que les données saisies depuis le mobile et depuis le web restent cohérentes en temps réel a nécessité un travail spécifique de conception des API, de gestion des conflits potentiels et de validation croisée.

- **Documentation et livrables (2 000 €)** : rédaction de la documentation technique (API, architecture, sécurité), de la documentation utilisateur avec captures d'écran annotées, et du dossier de réalisation professionnelle.

- **Gestion de projet et coordination (1 500 €)** : planification, suivi Jira, coordination avec le développement web pour les API partagées, gestion des priorités entre les deux projets.

Le projet a été mené de janvier à mars 2026, en parallèle de la formation, avec les phases suivantes : analyse des besoins mobiles et conception des API, développement itératif des écrans par rôle, tests fonctionnels et de synchronisation, et rédaction de la documentation.

Poste	Montant
Développement API REST (PHP/JSON)	5 000 €
Développement application Android (Java)	8 000 €
Maintenance future assurée sur 2 ans	3 000 €
Synchronisation et cohérence des données	2 500 €
Documentation et livrables	2 000 €
Gestion de projet et coordination	1 500 €
TOTAL	22 000 € TTC

4. Conception des données

4.1 Architecture client-serveur et API REST

L'application s'inscrit dans une architecture client-serveur. Le client Android (Java/Volley) envoie des requêtes HTTP au serveur (PHP/MySQL) via des API REST. Le serveur traite les requêtes, interroge la base et renvoie du JSON. Cette architecture centralise la logique métier côté serveur, partagée avec l'application web.

Les API REST couvrent : authentification, tableau de bord, CRUD comptes-rendus, praticiens, médicaments, utilisateurs et échantillons.

4.2 Structure du projet Android

- Activités : LoginActivity, DashboardActivity, SaisieCRActivity, ConsultationCRActivity, DetailCRActivity, RessourcesActivity, ProfilActivity, CRRegionActivity...
- Adaptateurs : RecyclerView adapters pour les listes
- Modèles : classes Java (Visiteur, CompteRendu, Praticien, Medicament)
- Layouts XML : Material Design (ConstraintLayout, CardView, Spinner, RecyclerView)

4.3 Base de données partagée (MCD)

L'application utilise la même base MySQL que l'application web. Le MCD (7 entités : VISITEUR, PRATICIEN, COMPTE_RENDU, MEDICAMENT, PRESENTER, OFFRIR, REGION) est décrit dans le dossier de la réalisation n°1 et n'a pas été modifié.

5. Gestion de projet

5.1 Méthodologie et outils

Développement individuel en parallèle de l'application web. Outils : Git/GitHub (https://github.com/Daifuku420/App_GSB_Mobile), Android Studio (Ladybug), Jira pour le suivi.

5.2 Suivi d'avancement avec Jira

Le tableau Jira du projet GSB a été utilisé pour les tâches mobiles (label « MOBILE »). Workflow : À faire → En cours → Terminé. Les tâches API communes sont tagguées « MOBILE + WEB ». La capture ci-dessous montre le tableau de bord Jira avec les tickets web et mobile.

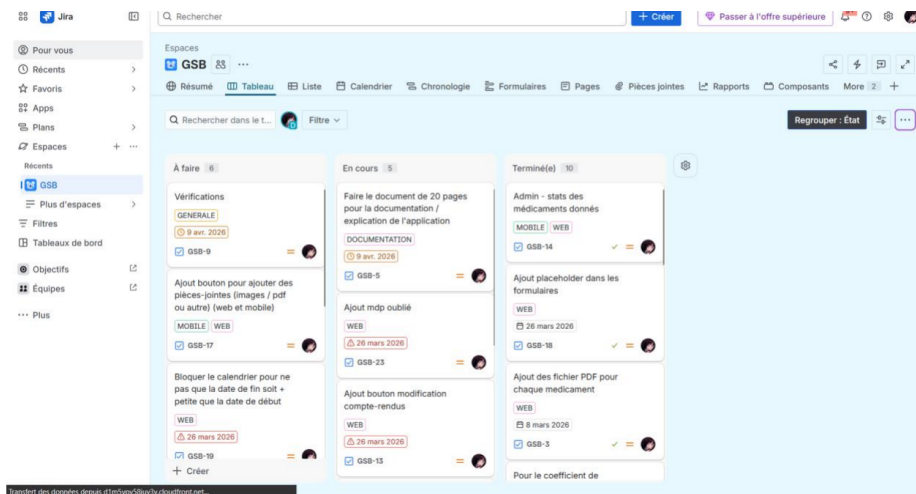


Figure 1 – Tableau Jira de suivi du projet GSB Appli-CR

Figure – Tableau Jira de suivi du projet GSB Appli-CR (tickets MOBILE et WEB)

6. Mode opératoire global de l'application

6.1 Rôle Visiteur

Navigation et tableau de bord

L'application dispose d'un menu latéral (drawer) accessible via l'icône hamburger en haut à gauche. Ce menu donne accès à l'ensemble des fonctionnalités : Accueil, Praticiens et Médicaments, Mes comptes-rendus, Créer un compte-rendu, et Comptes-rendus de ma région (pour les rôles supérieurs).

Le tableau de bord affiche quatre indicateurs clés sous forme de cartes : le nombre de comptes-rendus, le nombre de visites totales, le coefficient de notoriété moyen et la date de la dernière visite.

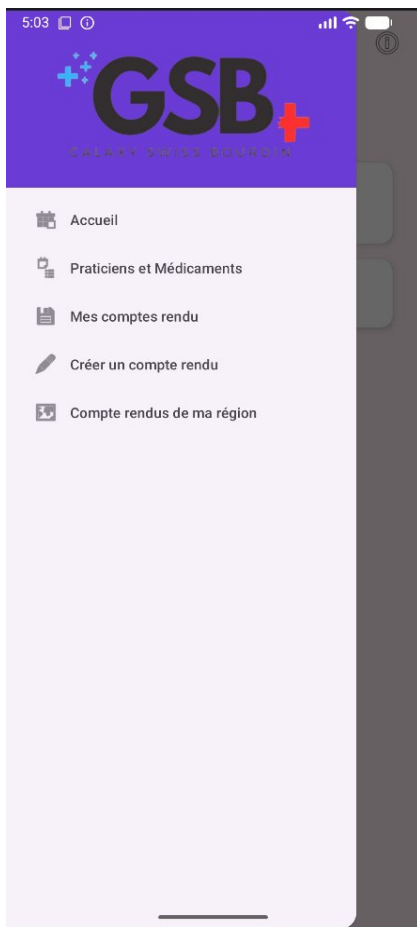


Figure 1 – Menu de navigation latéral

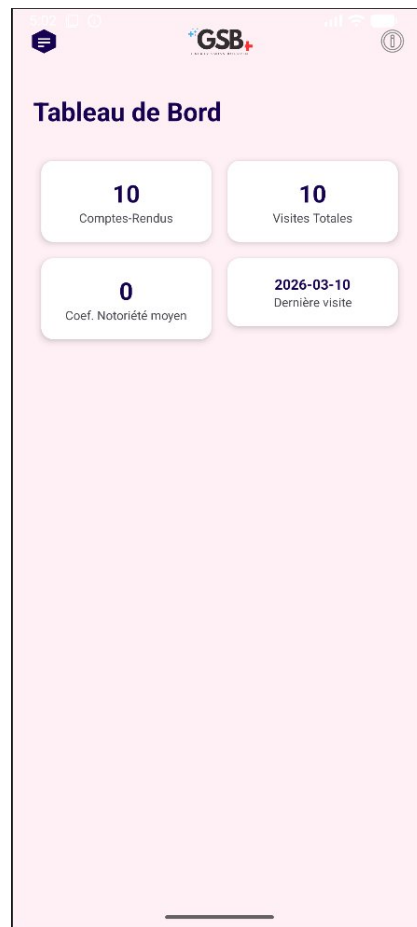


Figure 2 – Tableau de bord du visiteur

Saisie et consultation des comptes-rendus

L'écran de saisie permet de créer un nouveau compte-rendu en renseignant la date de visite, le praticien (Spinner), le motif, le médicament présenté (Spinner), une case à cocher pour l'échantillon, et le bilan textuel. Un bouton « Enregistrer le compte-rendu » valide la saisie.

L'historique des comptes-rendus s'affiche dans une liste scrollable, chaque élément indiquant la date et le motif. Un tap sur un élément ouvre le détail complet (numéro, date, praticien, motif, bilan) avec un bouton de retour.

Nouveau Compte-Rendu

Date de visite
AAAA-MM-JJ

Praticien
Grey Meredith

Motif
Ex: Visite annuelle

Médicament présenté
Advil 400

Donner un échantillon de ce médicament

Description / Bilan
Compte-rendu de la visite...

ENREGISTRER LE COMPTE-RENDU

Figure 3 – Formulaire de saisie d'un CR

Comptes Rendus - Région AL

2026-01-08	periodicite
2026-01-09	sollicitation
2026-01-08	remontage
2026-01-05	nouveaute
2026-01-02	remontage
2026-01-12	nouveaute
2026-03-10	sollicitation
2026-01-21	Nouveauté
2026-01-22	Nouveauté
2026-02-22	Visited annuelle
2026-02-22	

Figure 4 – Liste des comptes-rendus (région)

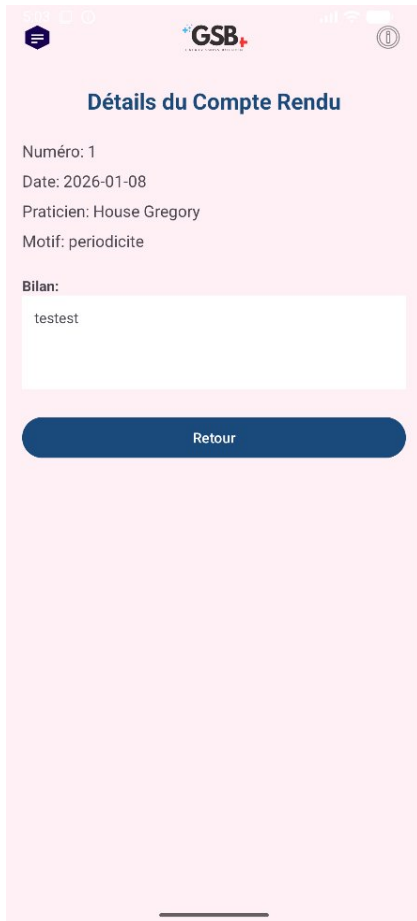


Figure 5 – Détail d'un compte-rendu

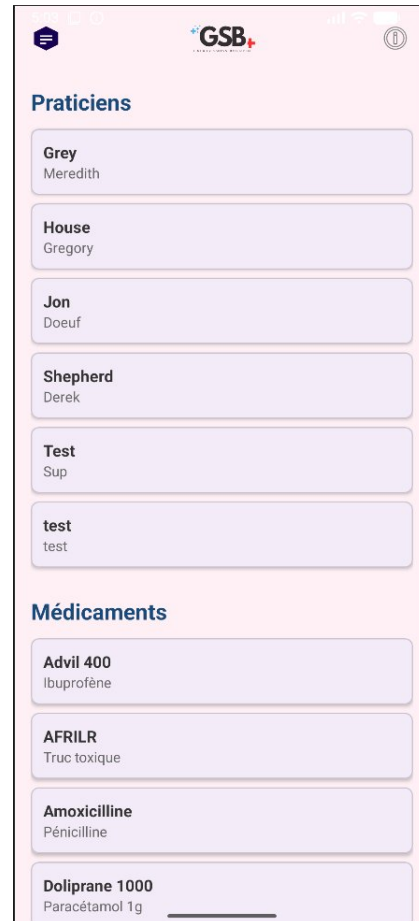


Figure 6 – Praticiens et médicaments

Ressources et profil

L'écran Ressources affiche la liste des praticiens (nom, prénom) et le catalogue des médicaments (nom, composition) dans des cartes distinctes. L'écran Mon Profil affiche les informations du visiteur connecté (prénom, nom, matricule, rôle avec région) et un bouton de déconnexion.



Figure 7 – Écran Mon Profil

6.2 Rôle Délégué régional

Le délégué régional dispose d'un menu « Comptes rendus de ma région » lui permettant de consulter l'ensemble des CR de sa région. L'affichage est identique à celui de la consultation personnelle, mais élargi à tous les visiteurs de la région.

7. Documentation technique

7.1 Environnement et technologies

Java pour Android, Android Studio Ladybug, SDK min 24 / cible 34. Volley pour les requêtes HTTP asynchrones. JSON pour l'échange de données. Back-end PHP 8 / MySQL partagé avec l'application web.

7.2 Communication avec l'API

Pattern : (1) StringRequest Volley avec paramètres POST → (2) API PHP → (3) requêtes PDO → (4) réponse JSON → (5) callback onResponse met à jour l'UI.

```
StringRequest request = new StringRequest(Request.Method.POST, API_URL,
    response -> { JSONArray arr = new JSONArray(response); /* traitement */ },
    error -> { /* gestion erreur */ }
) { @Override protected Map<String,String> getParams() {
    Map<String,String> p = new HashMap<>();
    p.put("matricule", matricule); return p;
}};
```

7.3 Gestion de l'authentification

LoginActivity envoie identifiant/mot de passe en POST. L'API PHP vérifie avec password_verify(). Succès : JSON (matricule, nom, prénom, rôle, région) stocké dans SharedPreferences. Déconnexion : effacement et retour à LoginActivity.

7.4 Sécurité

- Authentification obligatoire : vérification SharedPreferences sur chaque Activity
- Contrôle des rôles : écrans admin réservés aux rôles élevés
- Requêtes préparées PDO côté serveur (anti-injection SQL)
- Transmission POST (pas d'exposition dans les URL)
- Mots de passe hachés bcrypt (password_hash/password_verify)
- Protection XSS native Android

8. Tests et validation

Tests sur émulateur Android Studio (Pixel 7, API 34) et appareil physique.

- Connexion : identifiants valides/invalides, messages d'erreur
- Saisie CR : tous les cas (motifs variés, avec/sans échantillon)
- Synchronisation web ↔ mobile : saisie mobile → consultation web et inversement
- Consultation : cohérence des résultats avec l'application web
- Administration : CR de la région (délégué régional)
- Réseau : gestion timeout et erreurs Volley
- Navigation : transitions entre écrans, retour arrière, déconnexion

9. Bilan et évolutions futures

Le projet GSB Appli-CR Mobile a été mené à bien. L'ensemble des fonctionnalités a été développé et testé, offrant un outil de saisie mobile complet synchronisé avec l'application web.

Ce projet a été formateur : développement Android natif Java, communication Volley, parsing JSON, conception d'API REST PHP, gestion de deux projets interconnectés.

Suggestions d'améliorations

- Mode hors-ligne avec SQLite locale et synchronisation différée (complexité élevée)
- Notifications push pour les périodicités dépassées (complexité intermédiaire)
- Géolocalisation pour pré-remplir le praticien le plus proche (complexité intermédiaire)
- Module photo pour joindre des documents aux CR (complexité intermédiaire)
- Migration vers Kotlin (complexité faible à intermédiaire)
- Tests automatisés JUnit et Espresso (complexité intermédiaire)
- Pull-to-refresh sur les écrans de consultation (complexité faible)

10. Accès à l'application

Code source complet sur GitHub :

https://github.com/Daifuku420/App_GSB_Mobile

Identifiants de test

Rôle	Identifiant	Mot de passe
Visiteur médical	a11	Test2026
Délégué régional	d01	Japon2026
Responsable de secteur	r01	Japon2027

— Fin du dossier —

THOMAS Paul – BTS SIO SLAM – Session 2026