**Document de Spécifications - CONTROLsecuV2**

**1. Informations Générales**

**1.1 Identité du projet**

* **Nom de l'application : CONTROLsecuV2**
* **Version : 1.0**
* **Date de livraison cible : 01/07/2025**
* **Plateforme : Android (minSDK 30)**

**1.2 Description du projet**

**Application mobile destinée aux bailleurs sociaux pour effectuer des contrôles sur les éléments de sécurité des résidences. Cette application remplace entièrement l'ancienne version CONTROLsecu et fait partie d'un écosystème comprenant :**

* **Un logiciel de gestion complète du poste propreté**
* **CONTROLprop V2 (application de contrôle des prestations de propreté - déjà refaite)**
* **CONTROLsecuV2 (nouvelle version des contrôles de sécurité)**
* **Cohérence visuelle : Même design que CONTROLprop V2 pour uniformité de l'écosystème**

**1.3 Public cible**

* **Utilisateurs principaux : Professionnels des bailleurs sociaux**
* **Profil : Agents de terrain effectuant des contrôles de sécurité**
* **Contexte d'usage : Déplacements sur site, possibles interruptions de connexion**

**2. Exigences Fonctionnelles**

**2.1 Fonctionnalités Essentielles**

**2.1.1 Authentification (Login)**

* **Objectif : Sécuriser l'accès à l'application avec session unique**
* **Comportement :** 
  + **Écran de connexion avec identifiant/mot de passe**
  + **Case à cocher RGPD obligatoire : "J'ai lu et j'accepte les conditions de protection des données personnelles"**
  + **Lien vers conditions d'utilisation : Ouverture dans navigateur web**
  + **Validation RGPD : Connexion impossible sans acceptation des conditions**
  + **Sauvegarde des identifiants pour pré-remplissage automatique**
  + **Validation des credentials auprès du serveur**
  + **Réception d'une clé de session unique**
  + **Session unique : Un utilisateur ne peut être connecté que sur un seul appareil**
  + **Gestion des erreurs de connexion et session expirée**
  + **Déconnexion : Bouton disponible sur l'écran de sélection d'entrée**
* **Déblocage de session :** 
  + **Si session active ailleurs : Proposition de déblocage**
  + **Saisie obligatoire de l'email de contact**
  + **Envoi automatique d'un email de déblocage depuis le serveur**
  + **Écran d'attente avec message informatif**
  + **Possibilité de se connecter après validation du lien email**

**2.1.2 Sélection de l'entrée à contrôler**

* **Objectif : Permettre à l'utilisateur de naviguer dans la hiérarchie organisationnelle**
* **Structure hiérarchique :** 
  + **Client (= Organisme, niveau racine)**
  + **Agence (sous le client)**
  + **Groupement (sous l'agence)**
  + **Entrée (niveau final à contrôler)**
* **Gestion des droits :** 
  + **Droits définis côté serveur selon le client ET le niveau utilisateur**
  + **Filtrage automatique des données accessibles**
  + **Connexion obligatoire pour la première récupération**
  + **Cache local des données autorisées pour navigation hors ligne**
* **Comportement :** 
  + **Navigation arborescente à 4 niveaux**
  + **Icônes spécifiques pour chaque niveau (Agence, Groupement, Entrée)**
  + **Fil d'Ariane pour indiquer la position dans la hiérarchie**
  + **Recherche/filtrage à chaque niveau**
  + **Sauvegarde locale après première synchronisation**
  + **Navigation hors ligne une fois les données récupérées**

**2.1.3 Sélection du type de contrôle**

* **Objectif : Définir la périodicité du contrôle pour filtrer la grille**
* **Types disponibles :** 
  + **Mensuel (1) - Points à contrôler chaque mois**
  + **Trimestriel (2) - Points à contrôler tous les 3 mois**
  + **Semestriel (3) - Points à contrôler tous les 6 mois**
  + **Annuel (4) - Points à contrôler chaque année**
* **Comportement :** 
  + **Sélection obligatoire du type avant accès à la grille**
  + **Filtrage automatique des points selon le type sélectionné**
  + **Affichage du nombre de points concernés par type**

**2.1.4 Grille de contrôle**

* **Objectif : Interface principale pour effectuer le contrôle de sécurité**
* **Structure : Hiérarchie à 2 niveaux (Catégories → Points de contrôle)** 
  + **Catégories : Nombre variable selon le client (exemple : 9 sections)**
  + **Points de contrôle : Filtrés selon le type de contrôle sélectionné**
* **Statuts disponibles : Conforme / Non-conforme / N/A**
* **Gestion de l'urgence :** 
  + **Niveau d'urgence modifiable par le contrôleur (1-4)**
  + **Valeur par défaut définie dans la grille client**
* **Comportement :** 
  + **Affichage de la grille spécifique au client (format JSON)**
  + **Navigation par catégories avec icônes fournies**
  + **Saisie par point de contrôle avec les 3 statuts**
  + **Affichage des conseils/descriptions ("tip") pour chaque point**
  + **Modification possible du niveau d'urgence**
  + **Sauvegarde automatique locale en cas de perte de connexion**

**2.1.5 Clôture du contrôle**

* **Objectif : Finaliser et transmettre le contrôle effectué**
* **Options de clôture :** 
  + **Signature électronique : Optionnelle, proposée mais non obligatoire**
  + **Envoi immédiat : Optionnel, proposé mais non obligatoire**
  + **Sauvegarde locale : Automatique si pas de signature/envoi**
* **Comportement :** 
  + **Écran de choix : Signer et envoyer / Sauvegarder localement / Continuer plus tard**
  + **Signature électronique par tracé libre sur écran tactile (si choisi)**
  + **Génération automatique du rapport (gérée côté serveur)**
  + **Envoi du contrôle avec toutes ses données (si connexion + choix utilisateur)**
  + **Mise en file d'attente si hors ligne avec synchronisation automatique**
  + **Retour automatique : Vers la sélection d'entrée après clôture**

**2.2 Fonctionnalités Secondaires**

**2.2.1 Gestion des commentaires et photos**

* **Objectif : Documenter les anomalies constatées**
* **Comportement automatique :** 
  + **Ouverture automatique d'une proposition de commentaire lors d'une anomalie**
  + **Commentaire pré-rempli que l'utilisateur doit confirmer pour refuser**
  + **Prise d'une photo par anomalie via l'appareil photo (optionnelle)**
  + **Association photo/commentaire au point de contrôle spécifique**
  + **Compression/redimensionnement automatique des images (optimisé pour affichage 150px de largeur)**
  + **Prévisualisation de la photo avant validation**

**3. Exigences Techniques**

**3.1 Architecture et Technologies**

* **Langage : Kotlin**
* **IDE : Android Studio**
* **Architecture : MVVM (recommandé)**
* **Base de données locale : Room Database**
* **Communication réseau : Retrofit + OkHttp**
* **Logging et Analytics :** 
  + **Debug : Logs Android standard (Logcat)**
  + **Production : Firebase Crashlytics + Firebase Analytics**
  + **Système unifié : Interface unique s'adaptant automatiquement selon BuildConfig.DEBUG**

**3.2 Gestion des données**

**3.2.1 Stockage local**

* **Identifiants de connexion : Chiffrement sécurisé (SharedPreferences chiffrées)**
* **Contrôles en attente : Base de données locale avec statut de synchronisation**
* **Grilles de contrôle : Cache local pour fonctionnement hors ligne (structure dynamique selon client)**
* **Photos : Stockage temporaire local avant synchronisation**

**3.2.2 Synchronisation serveur**

* **Données à récupérer :** 
  + **Grilles de contrôle spécifiques au client (nombre et contenu variables)**
  + **Liste du patrimoine par utilisateur**
  + **Mises à jour des référentiels**
* **Données à envoyer :** 
  + **Contrôles terminés avec signatures**
  + **Photos et commentaires associés**
  + **Rapports générés**

**3.3 Gestion de la connectivité**

* **Mode connecté : Synchronisation en temps réel**
* **Mode déconnecté :** 
  + **Sauvegarde locale automatique**
  + **Queue de synchronisation**
  + **Reprise automatique à la reconnexion**
  + **Indicateur visuel du statut de connexion**

**4. Exigences Non-Fonctionnelles**

**4.1 Performance**

* **Temps de démarrage : < 3 secondes**
* **Fluidité : 60 FPS sur les écrans principaux**
* **Taille de l'APK : Optimisée au mieux (pas de limite stricte)**

**4.2 Sécurité**

* **Chiffrement : Données sensibles chiffrées localement**
* **Validation session : Test de session à chaque communication avec le serveur**
* **Transmission : HTTPS obligatoire pour toutes les communications**

**4.3 Utilisabilité**

* **Interface : Inspiration du design de CONTROLprop existant pour cohérence**
* **Charte graphique : Couleur principale #F17700 (orange vif)**
* **Style : Material Design adapté à l'écosystème**
* **Accessibilité : Support des fonctionnalités d'accessibilité Android**
* **Mode portrait/paysage : Adaptation automatique**

**5. Contraintes et Limitations**

**5.1 Contraintes techniques**

* **SDK minimum : Android API 30**
* **Charte graphique : Couleur principale #F17700, icônes disponibles en version orange (#F17700) et blanche (#FFFFFF)**
* **Icônes : Format PNG, style line art, disponibles en deux versions (orange/blanc), définition uniforme**
* **Firebase intégration : Crashlytics et Analytics pour la production**
* **Conformité RGPD : Case à cocher obligatoire + lien vers conditions d'utilisation**
* **Permissions requises :** 
  + **CAMERA (pour les photos)**
  + **INTERNET (synchronisation + Firebase + ouverture conditions d'utilisation)**
  + **WRITE\_EXTERNAL\_STORAGE (sauvegarde photos)**
  + **ACCESS\_NETWORK\_STATE (détection connectivité)**

**5.2 Contraintes projet**

* **Délai : Livraison impérative au 01/07/2025**
* **Compatibilité : Intégration avec l'écosystème existant**
* **Maintenance : Code documenté pour maintenance future**

**6. Écrans et Navigation**

**6.1 Architecture des écrans**

1. **Écran de connexion + Écran de déblocage session (saisie email + attente)**
2. **Écran de navigation hiérarchique (Client → Agence → Groupement → Entrée) + Bouton déconnexion**
3. **Écran de sélection du type de contrôle (Mensuel/Trimestriel/Semestriel/Annuel)**
4. **Écran de grille de contrôle (filtrée selon le type)**
5. **Écran de détail anomalie (commentaire automatique + photo optionnelle)**
6. **Écran de clôture (choix signature/envoi/sauvegarde)**
7. **Écran de signature (si choisi)**
8. **Retour automatique à l'écran 2 (sélection entrée)**

**6.2 Navigation**

* **Navigation principale : Navigation Drawer ou Bottom Navigation**
* **Retour en arrière : Sauvegarde automatique des données saisies**
* **État de synchronisation : Indicateur permanent visible**

**7. API et Intégrations**

**7.1 Endpoints requis**

* **POST /auth/login : Authentification + récupération des droits utilisateur + clé de session**
* **POST /auth/logout : Déconnexion et invalidation de la session**
* **POST /auth/unlock-request : Demande de déblocage de session (email requis)**
* **GET /auth/unlock-status/{requestId} : Vérification du statut de déblocage**
* **GET /hierarchy/{userId} : Hiérarchie complète accessible à l'utilisateur (filtrée selon droits)**
* **GET /grids/{clientId} : Récupération grilles de contrôle**
* **POST /controls : Envoi des contrôles terminés**
* **POST /controls/{id}/photos : Upload des photos**

**Authentification requise : Tous les appels API doivent inclure l'ID utilisateur + clé de session**

**7.2 Format des données**

* **Format d'échange : JSON**
* **Structure grille : Format hiérarchique (exemple fourni)**

**{**

**"categoryIndex": {**

**"id": number,**

**"txt": "Nom catégorie",**

**"ico": "icon\_id",**

**"pts": {**

**"pointIndex": {**

**"id": number,**

**"txt": "Nom point contrôle",**

**"tip": "Description/conseil",**

**"cat": "1234", // Types: 1=Mensuel, 2=Trimestriel, 3=Semestriel, 4=Annuel**

**"urg": "1-4" // Niveau urgence prédéfini par le client, modifiable par contrôleur**

**}**

**}**

**}**

**}**

* **Statuts de contrôle : "conforme" | "non-conforme" | "na"**
* **Authentification : JWT Token**
* **RGPD : URL des conditions d'utilisation dans la configuration**
* **Gestion d'erreurs : Codes HTTP standards + messages personnalisés**

**8. Tests et Validation**

**8.1 Tests fonctionnels**

* **Tests unitaires (> 80% couverture)**
* **Tests d'intégration API**
* **Tests UI automatisés**
* **Tests de synchronisation hors ligne**

**8.2 Tests utilisateurs**

* **Tests utilisabilité avec des agents terrain**
* **Validation des workflows métier**
* **Tests en conditions réelles (réseau instable)**

**9. Assets et Ressources**

**9.1 Icônes**

* **Icône application : Cône de chantier orange (#F17700)**
* **Icônes catégories : 19 icônes (ID 0-18) disponibles en orange et blanc**
* **Icônes navigation : Agence, Groupement, Entrée (pour la hiérarchie organisationnelle)**
* **Repository : https://github.com/DonBeny/CONTROLsecuV2/tree/main**
* **Mapping : Correspondance flexible selon les clients**

**9.2 Design et Interface (basé sur CONTROLprop)**

* **Palette de couleurs :** 
  + **Principal : #F17700 (orange vif)**
  + **Dégradé : Orange vers beige/crème**
  + **Texte : Gris foncé sur fond clair**
  + **Accents : Rose/magenta pour éléments interactifs**
* **Style d'interface :** 
  + **Header orange avec dégradé + titre centré + icône paramètres**
  + **Cartes/sections avec icônes orange alignées à gauche**
  + **Boutons arrondis orange pour les actions principales**
  + **Navigation avec fil d'Ariane et flèche retour**
  + **Champ de recherche intégré avec icône**
  + **Police claire avec hiérarchie typographique définie**

**10. Spécifications Détaillées des Écrans**

**10.1 Structure de la grille de contrôle**

**Architecture dynamique :**

* **Nombre de catégories : Variable selon le client (pas de limite fixe)**
* **Contenu des catégories : Spécifique à chaque client**
* **Points de contrôle : Filtrés selon le type sélectionné (cat: "1", "2", "3", "4")**

**Types de contrôle :**

* **Mensuel (1) : Points à vérifier chaque mois**
* **Trimestriel (2) : Points à vérifier tous les 3 mois**
* **Semestriel (3) : Points à vérifier tous les 6 mois**
* **Annuel (4) : Points à vérifier chaque année**

**Interface de contrôle :**

* **Sélection type : Écran dédié avant accès à la grille**
* **Filtrage automatique : Seuls les points correspondant au type sont affichés**
* **Navigation : Par catégories avec compteur de progression**
* **Adaptation automatique : Interface s'adapte au nombre de catégories**
* **Statuts disponibles : Conforme / Non-conforme / N/A (3 boutons/options)**
* **Gestion urgence : Possibilité de modifier le niveau (1-4) défini par défaut**
* **Aide contextuelle : Affichage des "tips" pour chaque point**
* **Anomalies : Bouton d'ajout commentaire + 1 photo pour les "Non-conforme"**
* **Sauvegarde : Indicateur visuel de l'état de sauvegarde (local/synchronisé)**
* **Icônes : PNG orange (#F17700) pour éléments actifs, blanc (#FFFFFF) pour contraste**
* **Flexibilité icônes : Correspondance ID→icône non contractuelle, laissée à l'appréciation du client**

**Exemple de structure (client type) :**

***Note : Cette structure est un exemple, chaque client aura sa propre configuration***

1. **Espaces extérieurs (ico: 12) - Colonnes sèches, coffrets, façades, luminaires, grilles EP**
2. **Locaux divers (ico: 7) - Portes d'accès, éclairages**
3. **Hall (ico: 2) - Plans sécurité, blocs secours, désenfumage, gaines, vidéosurveillance**
4. **Ascenseurs (ico: 3) - Portes, miroirs, éclairages cabine**
5. **Escaliers - Paliers (ico: 4) - Blocs secours, rambardes, colonnes sèches, gaines**
6. **Escaliers secours (ico: 4) - Éclairages, blocs secours, rambardes, exutoires**
7. **Sous-sol - Caves (ico: 8) - Portes, éclairages, gaines techniques**
8. **Parkings (ico: 11) - Plans sécurité, extincteurs, portes automatiques**
9. **Contrôles périodiques (ico: 0) - Vérifications prestataires obligatoires**

**10.2 Workflow de signature**

* **Écran dédié : Zone de tracé libre plein écran**
* **Contrôles : Boutons "Effacer" et "Valider"**
* **Validation : Vérification présence signature avant finalisation**

**11. Livraison et Déploiement**

**11.1 Livrables**

* **APK signé pour distribution**
* **Documentation technique**
* **Guide utilisateur**
* **Code source avec documentation**

**12. Navigation Hiérarchique Organisationnelle**

**12.1 Structure des données**

**{**

**"agences": [**

**{**

**"id": number,**

**"nom": "Nom agence",**

**"groupements": [**

**{**

**"id": number,**

**"nom": "Nom groupement",**

**"residences": [**

**{**

**"id": number,**

**"nom": "Nom résidence",**

**"adresse": "Adresse complète"**

**}**

**]**

**}**

**]**

**}**

**]**

**}**

**12.2 Interface de navigation**

* **Écran unique avec navigation par niveaux**
* **Fil d'Ariane : Organisme > Agence > Groupement > [Résidence]**
* **Icônes distinctes : Une icône spécifique pour Agence, Groupement, Résidence**
* **Compteurs : Affichage du nombre d'éléments à chaque niveau**
* **Recherche globale : Possibilité de rechercher directement une résidence**
* **Navigation retour : Boutons pour remonter dans la hiérarchie**

**14. Gestion des Anomalies et UX**

**14.1 Détection automatique d'anomalie**

* **Déclenchement : Statut "Non-conforme" sélectionné**
* **Proposition automatique : Commentaire pré-rempli basé sur le point de contrôle**
* **Confirmation requise : L'utilisateur doit explicitement refuser le commentaire proposé**
* **Photo optionnelle : Possibilité d'ajouter une photo**
* **Validation : Sauvegarde automatique du commentaire (sauf refus explicite)**

**14.2 Workflow de clôture flexible**

* **Options proposées :** 
  + **Signer et envoyer immédiatement**
  + **Sauvegarder localement (envoi différé)**
  + **Continuer plus tard (brouillon)**
* **Retour utilisateur : Après clôture, retour automatique à la sélection d'entrée**
* **Navigation cyclique : Permet d'enchaîner plusieurs contrôles facilement**

**16. Système de Logging et Analytics**

**16.1 Architecture du logging**

* **Interface unifiée : Système de logging unique s'adaptant automatiquement à l'environnement**
* **Debug mode : Logs Android standard via Logcat pour le développement**
* **Production mode : Firebase Crashlytics + Analytics pour le monitoring en production**
* **Commutation automatique : Basée sur BuildConfig.DEBUG**

**16.2 Fonctionnalités de logging**

* **Niveaux standards : VERBOSE, DEBUG, INFO, WARN, ERROR, FATAL**
* **Tracking utilisateur : Actions, navigation entre écrans, erreurs**
* **Propriétés utilisateur : ID utilisateur, client, rôle**
* **Événements métier : Contrôles effectués, anomalies détectées, synchronisations**

**16.3 Firebase Analytics Events**

* **user\_action : Actions utilisateur (login, sélection, validation)**
* **screen\_view : Navigation entre écrans**
* **control\_completed : Contrôle terminé avec métadonnées**
* **sync\_event : Synchronisations réussies/échouées**
* **error\_occurred : Erreurs applicatives avec contexte**

**16.4 Crashlytics Integration**

* **Crash reporting : Automatique pour tous les crashes**
* **Custom logging : Logs personnalisés avant les crashes**
* **User identification : Association crashes ↔ utilisateurs**
* **Breadcrumbs : Historique des actions avant erreur**

**18. Conformité RGPD et Conditions d'Utilisation**

**18.1 Interface de consentement**

* **Case à cocher obligatoire : "J'ai lu et j'accepte les conditions de protection des données personnelles"**
* **État par défaut : Non cochée (l'utilisateur doit cocher explicitement)**
* **Validation : Impossible de se connecter sans acceptation**
* **Lien cliquable : Texte avec lien vers les conditions d'utilisation**

**18.2 Ouverture des conditions**

* **Action : Clic sur le lien dans le texte RGPD**
* **Comportement : Ouverture dans le navigateur par défaut**
* **URL : Configurable dans les paramètres de l'application**
* **Retour : L'application reste active en arrière-plan**

**18.3 Stockage du consentement**

* **Sauvegarde locale : Date et heure de l'acceptation**
* **Transmission serveur : Horodatage du consentement avec les données de connexion**
* **Révocation : Possibilité de révoquer depuis les paramètres de l'application**

**18.4 Configuration**

**{**

**"rgpd": {**

**"terms\_url": "https://www.orgaprop.org/protectDonneesPersonnelles.html",**

**"privacy\_url": "https://www.orgaprop.org/protectDonneesPersonnelles.html",**

**"consent\_text": "J'ai lu et j'accepte les conditions de protection des données personnelles",**

**"required": true**

**}**

**}**

**20. Gestion des Droits Utilisateur**

**20.1 Niveaux d'accès**

* **Abonnement client : Le client doit être abonné au module CONTROLsecu**
* **Niveau 1 : Contrôles uniquement sur les entrées spécifiquement attribuées à l'utilisateur**
* **Niveau 2 : Contrôles sur toute l'agence de rattachement**
* **Niveau 3+ : Contrôles sur toutes les agences du client**

**20.2 Filtrage des données**

* **Côté serveur : Filtrage automatique selon abonnement + niveau utilisateur**
* **Cache local : Seules les données autorisées sont téléchargées**
* **Interface : L'utilisateur ne voit que ce à quoi il a accès**
* **Validation : Contrôle des droits à chaque action critique**

**21. Livraison et Déploiement**

**21.1 Livrables**

* **APK signé pour distribution**
* **Documentation technique**
* **Guide utilisateur**
* **Code source avec documentation**

**21.2 Distribution**

* **Distribution interne aux clients**
* **Mécanisme de mise à jour à prévoir**