📘 Cahier des Charges Phase 1 — Version Mise à Jour avec Jira / Confluence

1. Contexte du Projet

Le projet vise à développer une plateforme web intelligente capable de transformer les avis utilisateurs du Google Play Store en données exploitables. L'objectif est d'optimiser les produits et améliorer l'expérience utilisateur via une analyse fine des retours.

Les utilisateurs disposeront d'un dashboard interactif et d'un chatbot conversationnel pour faciliter la prise de décision et le suivi des problèmes critiques.
Les incidents critiques pourront être automatiquement remontés dans Jira et certaines analyses ou rapports documentés dans Confluence, pour centraliser le suivi et la documentation.

---

2. Objectifs du Projet

- **Collecter automatiquement tous les avis Google Play des applications ciblées.**
- **Analyser les avis pour déterminer :**
  - **Sentiment (positif, négatif, neutre)**
  - **Catégorisation thématique des avis négatifs (bugs, lenteur, crash…)**
  - **Priorisation des problèmes critiques.**
- **Restituer l'information via :**
  - **Dashboard interactif avec KPIs et filtres**
  - **Chatbot conversationnel pour questions métier**
  - **Tickets automatiques dans Jira pour incidents critiques**
  - **Documentation automatique dans Confluence pour rapports et analyses**
- **Assurer une architecture modulaire, scalable et sécurisée.**

---

3. Périmètre Fonctionnel

3.1 Collecte d'avis

- **Ciblage uniquement Google Play Store**
- **Récupération des avis avec métadonnées : version, note, date, langue, localisation**
- **Pipeline automatique avec planification régulière**

3.2 Analyse IA / NLP

- **Détection du sentiment : positif, neutre, négatif**
- **Classification thématique des avis négatifs**
- **Extraction des mots-clés et tendances pour alertes et reporting**
- **Score de priorité pour la gestion des problèmes critiques**

3.3 Interface Utilisateur

- Dashboard avec KPIs et graphiques
- Filtres avancés : date, version, sentiment, thème
- Chatbot interactif pour interrogation en langage naturel
- Export CSV/PDF des données et rapports

3.4 Intégration Jira / Confluence

- Jira :
    - Création automatique de tickets pour problèmes critiques
    - Assignation selon règles de priorité
    - Statistiques et suivi via API
- Confluence :
    - Documentation automatique de certaines analyses et rapports
    - Historique et archivage des insights
    - Accès collaboratif aux documents générés

3.5 Fonctionnalités supplémentaires

- Gestion multi-applications
- Historique complet des avis et analyses
- Administration : fréquence scraping, seuils d'alerte, gestion des utilisateurs
- Monitoring plateforme : pipeline scraping, IA et API backend

---

4. Acteurs et Rôles

Rôle	Responsabilités
Chef de projet / Architecte	Supervision globale, validation des livrables, suivi du planning et risques
Lead Frontend	Design et validation UX/UI, interface dashboard/chatbot
Frontend Developer	Implémentation interface, visualisations, filtres
Lead Backend	Architecture API, base de données, sécurité, intégration Jira/Confluence
Backend Developer	Intégration IA, API, logique métier, automation Jira/Confluence
Scraping Engineer	Pipeline de collecte et nettoyage des avis Google Play
Data / NLP Engineer	Modèles IA pour sentiment et classification thématique

QA / Test Engineer	Tests fonctionnels, unitaires, intégration et sécurité
DevOps Engineer	CI/CD, déploiement, monitoring, scalabilité

---

5. Cas d'Usage et Besoins Métier

Cas d'usage	Description	Objectif
Sélection application	Choix de l'app à analyser	Déterminer cible des analyses
Paramétrage collecte	Période, volume d'avis, versions	Contrôle du volume et granularité
Analyse de sentiment	Voir répartition P/Neutre/N	Identifier points forts et axes d'amélioration
Classification thématique	Organisation avis négatifs par type de problème	Prioriser corrections et interventions
Tableau de bord KPI	Visualisation des tendances, top problèmes, note moyenne	Aide à la décision
Chatbot conversationnel	Questions en langage naturel	Réponse rapide et synthétique aux besoins métiers
Création tickets Jira	Génération de tâches pour problèmes critiques	Traçabilité et action rapide
Documentation Confluence	Archivage des analyses et rapports	Partage et historique des insights
Export / reporting	Rapports PDF ou CSV	Partage des résultats avec équipes et direction
Monitoring plateforme	Suivi état scraping, IA et API	Assurer disponibilité et performance

---

6. Contraintes et Exigences Techniques

- Backend : Python (FastAPI) ou Node.js (Express/Nest)
- Frontend : Next.js + Tailwind + Shadcn UI
- Base de données : PostgreSQL ou MongoDB
- IA / NLP : HuggingFace Transformers, PyTorch, scikit-learn
- Scraping : Python + Scrapy / google-play-scraper, Playwright si nécessaire
- Intégration Jira / Confluence : API REST Atlassian
- Sécurité : Auth JWT/OAuth, protection API, monitoring Sentry/OWASP ZAP

- **Scalabilité / DevOps : Docker, Docker Compose, CI/CD GitHub Actions, monitoring Grafana/Prometheus**