

# Projet – Flutter

## Application de scoring de crédit éthique basée sur l'IA explicable (XAI)

Repo github : <https://github.com/feirouzzzz/CreditXAI.git>

### I. Figma

<https://www.figma.com/files/team/1563163384276202092/project/480845521/Team-project?fuid=1576528221857941756>

### II. Product Backlog

⭐ ÉPIC 1 — Modèle IA (ML + SHAP + Fairness)

🎯 Objectif : Construire un modèle éthique, fiable, explicable.

ID	User Story	Priorité
US1.1	Collecter + préparer un dataset propre	Haute
US1.2	Entraîner un modèle ML fiable	Haute
US1.3	Obtenir un score précis pour l'utilisateur	Haute
US1.4	Comprendre la prédiction via SHAP	Haute
US1.5	Mesurer et réduire les biais (fairness)	Haute

✓ Tâches

(Assignées à M1 : Driss — ML Engineer)

- Collecte dataset
- Nettoyage, encoding, normalisation
- Train/test split
- Entraînement Logistic Regression + RandomForest

- Export modèle (.joblib)
- SHAP : force\_plot, summary\_plot
- Fairness (disparate impact, equal opportunity...)
- Tests unitaires ML
- Documentation ML

★ ÉPIC 2 — Backend FastAPI & API Sécurisée

⌚ Objectif : API robuste, sécurisée, documentée.

ID	User Story	Priorité
US2.1	Endpoint /score	Haute
US2.2	Endpoint /explain	Haute
US2.3	Logging audit des requêtes	Haute
US2.4	Authentification JWT	Moyenne
US2.5	Documentation Swagger/Redoc	Moyenne

✓ Tâches

(Assignées à M2 : Soulaiman — Backend)

- Structure FastAPI
- Endpoint /score (POST)
- Endpoint /explain (POST)
- Endpoint /health (GET)
- Connexion PostgreSQL
- Tables : logs + users
- Middleware logging
- JWT Auth
- Swagger / Redoc
- Tests Pytest

- Documentation backend

★ ÉPIC 3 — Application Mobile Flutter

⌚ Objectif : UI intuitive + intégration complète API.

ID	User Story	Priorité
US3.1	Formulaire crédit	Haute
US3.2	Affichage score	Haute
US3.3	Affichage SHAP simplifié	Haute
US3.4	Dashboard analyste	Moyenne

✓ Tâches

(Assignées à M3 : Zakaria — Flutter)

- Setup projet Flutter
- UI formulaire + validation
- Appels API /score /explain
- Affichage du score (Card)
- Graphiques SHAP (bar chart ou radar)
- Dashboard analyste (optionnel)
- Stockage local (logs)
- Tests unitaires Flutter
- Documentation Flutter

★ ÉPIC 4 — DevOps, Docker & CI/CD, Testing & Qualité Continue

⌚ Objectif : Construire un pipeline DevOps complet incluant CI/CD, conteneurisation, tests automatisés (Selenium, JMeter), qualité du code (SonarQube) et monitoring.

(Assignées à M4: Feirouz — DevOps & automatisation)

## ● US4.1 — Conteneurisation Backend & Environnement Dev

★ En tant que DevOps, je veux conteneuriser tout le backend afin d'avoir un environnement stable et reproductible.

### ✓ Tâches

- Créer Dockerfile Backend
- Créer Docker-compose (Backend + PostgreSQL + ML Service)
- Gestion images Docker (clean/tag/push)
- Monitoring des conteneurs (logs + santé)
- Documentation installation Docker

## ● US4.2 — Pipeline CI/CD Automatisé (GitHub Actions)

★ En tant que DevOps, je veux automatiser la construction, les tests et le déploiement du projet.

### ✓ Tâches

#### Build & tests

- Pipeline : Build → Tests → Analyse Qualité → Build Docker → Deploy
- Tests unitaires backend
- Tests ML
- Tests Flutter
- Génération rapports JUnit/HTML
- Condition : déploiement seulement si 100% des tests réussissent

#### Build App Mobile

- Workflow GitHub Actions pour générer automatiquement l'APK Flutter
- Artifact automatique sur GitHub Releases

#### Déploiement Cloud

- Deployment automatique sur Railway / Render / AWS
- Configuration variables d'environnement

- Vérification service health checks

#### ● US4.3 — Qualité du Code & Analyse Statique (SonarQube)

★ En tant que DevOps, je veux analyser en continu la qualité du code pour éviter les bugs et dettes techniques.

##### ✓ Tâches

- Installer SonarQube / héberger SonarQube
- Installer SonarScanner
- Créer fichier sonar-project.properties
- Intégration GitHub Actions (analyse automatique à chaque push)
- Configuration Quality Gates
- Rapport automatique + badges qualité
- Correction des anomalies détectées

#### ● US4.4 — Tests UI Automatisés (Selenium)

★ En tant que DevOps, je veux automatiser la validation de l'interface utilisateur.

##### ✓ Tâches

- Installer Selenium WebDriver
- Implémenter Page Object Model
- Scénarios UI automatisés :
  - login
  - affichage score crédit
  - visualisation SHAP
- Exécution cross-browser : Chrome + Firefox
- Intégration CI/CD avec GitHub Actions
- Génération rapport HTML / Allure

- Notifications en cas d'erreur

### ● US4.5 — Tests de Performance & Charge (JMeter)

★ En tant que DevOps, je veux valider les performances et la stabilité du système sous charge.

#### ✓ Tâches

- Installation JMeter
- Création des plans de test (.jmx) :
  - API scoring
  - API explainability (SHAP)
  - stress test
  - load test
  - spike test
- Exécution automatique dans GitHub Actions
- Génération rapports HTML + graphes
- Analyse résultats
- Optimisation backend + base de données

### ● US4.6 — Centralisation des Tests & Monitoring Qualité

★ En tant que DevOps, je veux centraliser tous les résultats pour avoir un tableau de bord complet de la qualité.

#### ✓ Tâches

- Installation & configuration Allure Reports
- Dashboard qualité (UI Tests + Performance + Qualité Code)
- Centralisation logs Selenium
- Centralisation rapports JMeter
- KPIs :

- taux de succès des tests
- temps de réponse backend
- couverture SonarQube
- disponibilité des services
- Notifications Slack / Teams
- Documentation DevOps + slides soutenance + vidéo

### III. Répartition finale des tâches (propre & lisible)

Membre	Rôle	Responsabilités
<b>Driss (M1)</b>	ML Engineer	Data + Modèle IA + SHAP + Fairness
<b>Soulaiman (M2)</b>	Backend Developer	API FastAPI + DB + sécurité
<b>Zakaria (M3)</b>	Flutter Developer	App mobile + UI/UX + intégration API
<b>Feirouz (M4)</b>	DevOps Engineer	Docker + CI/CD + Selenium + JMeter + SonarQube + Monitoring