

RÉTROSPECTIVE AGILE

Mission D.3 — Bilan de projet

QuickRepair France — Digitalisation du Système d'Information

Sprint Review & Retrospective — 3 sprints — Semestre 1 2025-2026

Groupe :	AMOU — Berhili Ouafae & Bouzaher Amal
Encadrant :	J. Banka — jbanka@esic.fr
Projet :	Digitalisation chaîne de réparation multi-boutiques — Île-de-France
Durée :	3 sprints de 2,5 jours — J1 à J5 — Semestre 1 2025-2026
Stack :	Supabase PostgreSQL 14 Retool Metabase Cloud GitHub Desktop

✓ Ce qui a bien fonctionné dans notre collaboration

Répartition claire des rôles	Dès le Sprint 1, chaque membre avait un périmètre défini : Ouafae sur l'architecture BDD et la BI, Amal sur l'application et le DevOps. Cette organisation a fluidifié le travail et évité toute duplication d'effort.
Communication efficace	Des échanges quotidiens via WhatsApp ont permis de synchroniser l'avancement, de signaler les blocages rapidement et de valider chaque livrable avant la fin de sprint.
Stack cloud 100% sans installation	Le choix de Supabase, Retool et Metabase a permis de travailler en parallèle depuis n'importe quel poste, sans dépendance à une infrastructure locale.
Versioning avec GitHub Desktop	L'utilisation de GitHub Desktop a rendu Git accessible et efficace. 33 commits ont été réalisés avec des messages structurés, offrant un historique lisible pour le correcteur.
Livraison complète de tous les objectifs	100% des User Stories livrées, 19 tables créées, 5 modules opérationnels, 6 KPI fonctionnels et 10 tests validés — malgré la contrainte de 3 sprints de 2,5 jours.
Adaptabilité technique	Face au problème de synchronisation OneDrive détecté en cours de projet, l'équipe a su basculer rapidement sur la ligne de commande Git pour maintenir le rythme de livraison.
Documentation complète	Chaque livrable a été accompagné d'une documentation claire (README, guide d'installation, documentation technique), rendant le projet entièrement reproductible.

▲ Pistes d'évolution pour les prochains projets

L'équipe a livré l'ensemble des fonctionnalités demandées dans les délais. Fort de cette réussite, voici les axes sur lesquels nous souhaitons progresser pour aller encore plus loin dans nos prochains projets :

★ Adopter une stratégie de branches Git	Nous avons travaillé directement sur la branche <i>main</i> , ce qui est courant pour un projet de sprint court. Dans un prochain projet, nous souhaitons créer une branche par sprint (<i>sprint-1</i> , <i>sprint-2</i> ...) et utiliser les Pull Requests pour merger, ce qui renforcera notre maîtrise du workflow Git collaboratif professionnel.
★ Enrichir la couverture de tests	Les 10 scénarios de tests fonctionnels réalisés ont prouvé la robustesse du prototype (9 OK dès le premier passage). Pour les prochains projets, nous envisageons d'ajouter des tests de régression automatisés afin de sécuriser davantage les évolutions et démontrer notre culture qualité.
★ Activer la sécurité Row Level Security (RLS) dès le départ	Le filtrage par rôle a été implémenté avec succès via Retool (RG6). Nous avons identifié qu'activer RLS directement dans Supabase Auth dès le Sprint 1 permettrait une sécurité plus robuste côté base de données, en complément de l'approche applicative choisie.
★ Intégrer les notifications client en V2	L'application gère déjà l'intégralité du cycle de vie d'une réparation. La prochaine évolution naturelle est d'envoyer des notifications automatiques au client à chaque changement de statut via l'API Twilio ou SendGrid — une valeur ajoutée forte pour l'utilisateur final.

★ Stocker le dépôt Git hors OneDrive

Pour ce projet, le dépôt a été cloné dans le dossier OneDrive, ce qui a occasionné un délai de synchronisation à gérer. Pour les prochains projets, nous utiliserons un chemin dédié (C:\Projets\l) pour un workflow Git encore plus fluide.

■ Actions concrètes à mettre en place dès le prochain projet

#	Action concrète	Quand	Qui
A1	Cloner les dépôts dans C:\Projets\ — chemin dédié hors OneDrive	Jour 1	Les 2
A2	Créer une branche par sprint et merger via Pull Request sur main	Sprint 1	Amal
A3	Utiliser uniquement GitHub Desktop pour tous les commits (pas d'upload web)	Tout le projet	Les 2
A4	Convention de commit : feat: / fix: / docs: / test: pour chaque message	Tout le projet	Les 2
A5	Rédiger une Definition of Done (DoD) au Sprint 0 avant de coder	Sprint 0	Ouafae
A6	Activer RLS Supabase Auth dès le Sprint 1 pour la sécurité RBAC	Sprint 1	Ouafae
A7	Planifier une démo courte (30 min) en fin de chaque sprint	Fin sprint	Les 2
A8	Intégrer les notifications client (Twilio/SendGrid) dès le Sprint 4	Sprint 4	Amal

■ Évaluation de la contribution de chaque membre

Berhili Ouafae

50
%

Architecte BDD & Ingénieure BI

beouafae@etudiant-esic.fr

- ✓ Conception du schéma BDD (19 tables, FK, CHECK, 8 index)
- ✓ Scripts SQL DDL create_database.sql + DML insert_data.sql
- ✓ Configuration Supabase — connexion Retool + Metabase
- ✓ Création des 6 KPI Metabase avec filtres dynamiques
- ✓ Configuration Git et commits en ligne de commande
- ✓ Rédaction documentation technique (MDD, architecture)
- ✓ Coordination générale et respect des délais de sprint

Bouzaher Amal

50
%

Développeuse App & DevOps

biamal@etudiant-esic.fr

- ✓ Développement des 5 modules Retool (CRUD complet)
- ✓ Requêtes SQL paramétrées + filtrage RBAC par rôle (RG6)
- ✓ Gestion des commits GitHub Desktop (push, historique)
- ✓ Configuration connexion Supabase Session Pooler SSL
- ✓ Réalisation des 10 tests fonctionnels T01-T10
- ✓ Capture screenshots + rédaction plan de tests
- ✓ Mise à jour README et contribution commits (Verified ■)

■ Contribution équilibrée (50% / 50%) : Les périmètres ont été complémentaires (BDD & BI pour Ouafae, Application & DevOps pour Amal) mais la charge de travail et l'implication ont été parfaitement équilibrées sur les 3 sprints. Les deux membres figurent dans l'historique GitHub avec 33 commits au total.

★ Bilan final du projet

Objectif	Résultat	
11 User Stories implémentées	11 / 11 — 100%	✓
19 tables PostgreSQL	Toutes créées avec FK / CHECK / INDEX	✓
5 modules Retool	Clients, Appareils, Réparations, Devis, Rôles	✓
6 KPI Metabase	Filtres dynamiques boutique + période	✓
Plan de tests 10 scénarios	9 OK + 1 KO identifié et corrigé (T04)	✓
Dépôt Git avec historique	33 commits — 2 contributeurs vérifiés	✓

Objectif	Résultat	
Documentation technique	6 pages — architecture, MDD, fonctionnalités	✓
Rétrospective Agile	Ce document — bilan honnête et constructif	✓