

Compte Rendu du Projet

Plateforme de surveillance des comptes bancaires

avec analyse des transactions et détection de fraude

Réalisé par : Bensitel Ilyasse
Bahou abdelhamid
Karrouach ansar
Charmaqe Hamza

Encadré par : Prof Sarah Aouhassi

Master Big Data et IoT - 1re Année

Gestion de Projet

28 avril 2025

Table des matières

Introduction Générale	3
I Cahier des Charges	4
1 Introduction	5
1.1 Présentation du projet	5
1.2 Enjeu	5
1.3 Bénéficiaires	5
2 Objectifs	6
2.1 Objectifs du projet	6
2.2 Livrables attendus	6
3 Spécifications et Exigences du Système	7
3.1 Exigences fonctionnelles	7
3.2 Exigences Non Fonctionnelles	7
3.3 Spécifications Techniques	8
3.3.1 Architecture	8
3.3.2 Technologies utilisées	8
II Organigramme des tâches (WBS)	9
4 Estimation de la durée	13
III Ordonnancement des tâches (Diagramme de Gantt)	14
5.1 Définition du diagramme de Gantt	15
5.2 Tableau d'Ordonnancement des Tâches	15
5.3 Diagramme de Gantt	16
IV Technique de PERT (Chemin critique)	17
6.1 Tableau des tâches avec durées et dépendances	18
6.2 Diagramme de PERT	18
V Management des risques et Matrice des Compétences	19
7.1 Matrice de gestion des risques	20
8.1 Présentation de la matrice	21

VI Estimation des coûts internes et externes	23
Conclusion Générale	26

Introduction Générale

Dans un monde où les transactions financières numériques ne cessent de croître, la fraude bancaire représente un défi majeur pour les institutions financières. Chaque année, des milliards d'euros sont perdus à travers le monde en raison d'activités frauduleuses toujours plus sophistiquées. Face à cette problématique, la mise en place de systèmes intelligents et automatisés de surveillance devient une nécessité.

Le présent projet s'inscrit dans cette dynamique en proposant le développement d'une plateforme de surveillance des comptes bancaires, basée sur les technologies du Big Data et du Machine Learning. Cette plateforme a pour objectif principal la détection proactive des anomalies et des fraudes en temps réel, tout en garantissant la confidentialité et la sécurité des données des utilisateurs.

Grâce à une architecture microservices, une analyse continue des transactions, et une interface de visualisation intuitive, la solution vise à offrir aux banques et à leurs clients un outil performant de prévention des risques. En adoptant une méthodologie agile, le projet est structuré autour de phases bien définies incluant l'analyse des besoins, la conception, le développement, les tests, et la mise en production.

Ce rapport présente l'état d'avancement du projet à travers une description complète du cahier des charges, des objectifs, des spécifications techniques et fonctionnelles, ainsi qu'un découpage détaillé des tâches via le WBS. Il constitue ainsi une base de référence pour le pilotage, le suivi et l'évaluation du projet.

Première partie

Cahier des Charges

1 Introduction

1.1 Présentation du projet

Le projet consiste à développer une plateforme intelligente de surveillance bancaire utilisant le Big Data et le Machine Learning pour la détection en temps réel des fraudes. Le système analysera les transactions, détectera les comportements suspects, et enverra des alertes instantanées tout en s'appuyant sur des technologies modernes pour assurer la sécurité des utilisateurs.

1.2 Enjeu

Ce projet est crucial pour lutter contre la fraude bancaire, un problème qui coûte des milliards de dollars chaque année. En permettant une surveillance continue et une détection rapide des activités frauduleuses, la plateforme protège les utilisateurs bancaires et les institutions financières. Le système sera particulièrement utile dans la prévention des fraudes sur les cartes bancaires, les transactions internationales et les opérations suspectes sur les comptes. En outre, il garantit une meilleure sécurité des données personnelles et bancaires des utilisateurs.

1.3 Bénéficiaires

- **Banques** : Les institutions financières bénéficieront de la réduction des risques liés à la fraude et à la protection de leurs clients.
- **Utilisateurs bancaires** : Les clients des banques verront leurs informations financières protégées et recevront des alertes en temps réel en cas de transactions suspectes.
- **Institutions financières** : Les régulateurs financiers auront un aperçu plus clair des comportements frauduleux, contribuant à un environnement financier plus sûr et transparent.

2 Objectifs

2.1 Objectifs du projet

- Développer un système de surveillance des comptes bancaires en temps réel
- Implémenter des algorithmes de machine learning pour la détection de fraudes
- Créer une interface utilisateur intuitive pour le suivi des transactions
- Mettre en place un système d'alertes instantanées
- Assurer la sécurité et la confidentialité des données

2.2 Livrables attendus

- Code source complet et documenté
- Documentation technique et d'utilisation
- Rapports de tests (unitaires, d'intégration, de performance)
- Manuel d'installation et de déploiement
- Procédures de maintenance et de sauvegarde

3 Spécifications et Exigences du Système

3.1 Exigences fonctionnelles

- **Analyse des transactions** : Le système doit être capable d'analyser toutes les transactions bancaires en temps réel, de détecter les anomalies et d'émettre des alertes.
- **Détection des fraudes** : Utilisation des algorithmes de Machine Learning pour analyser les comportements des utilisateurs et identifier toute activité frauduleuse.
- **Gestion des alertes** : Envoi d'alertes instantanées par email, SMS ou notifications push lorsqu'une fraude est détectée.
- **Tableaux de bord** : Les administrateurs auront accès à des tableaux de bord pour visualiser les alertes, les statistiques de transactions et l'historique des événements.

3.2 Exigences Non Fonctionnelles

- **Performance** : Le système doit être capable de traiter un grand volume de transactions par seconde (Big Data).
- **Disponibilité** : Le système doit être hautement disponible et avoir une tolérance aux pannes afin de garantir un service 24/7.
- **Scalabilité** : La plateforme doit être capable de supporter une augmentation du nombre d'utilisateurs et de transactions sans dégradation des performances.
- **Maintenance et support** : Le système doit inclure un mécanisme de mise à jour continue, ainsi qu'un support technique pour résoudre les incidents en temps réel.

3.3 Spécifications Techniques

3.3.1 Architecture

Le système sera basé sur une architecture microservices pour assurer la flexibilité, la scalabilité et la maintenance. Chaque service sera indépendant et interagira avec les autres via des API. Les principaux services incluent la gestion des transactions, la détection de fraude, l'envoi d'alertes et la gestion des utilisateurs.

3.3.2 Technologies utilisées

Catégorie	Technologies utilisées
Big Data	Hadoop, Spark
Machine Learning	Scikit-learn, TensorFlow
Bases de données	PostgreSQL ou MariaDB (pour les données transactionnelles)
Framework backend	Spring Boot
Framework frontend	React ou Angular (pour l'interface utilisateur)

TABLE 3.1 – Technologies utilisées dans le système

Deuxième partie

Organigramme des tâches (WBS)

1. Définition du WBS

Organigramme des Tâches du Projet (OTP), Structure de Découpage du Projet (SDP) ou encore Work Breakdown Structure (WBS) en anglais, est une subdivision hiérarchique des tâches qui doivent être réalisées pour atteindre les objectifs fixés pour un projet donné. Son but est de faciliter l'organisation du projet en définissant l'ensemble de son contenu, en fournissant une référence pour la planification des activités et l'établissement du budget prévisionnel. Elle peut également être utilisée pour le management des risques, l'identification des acquisitions nécessaires, et la délégation des missions confiées à chaque acteur impliqué dans le projet.

2. Illustration de l'WBS

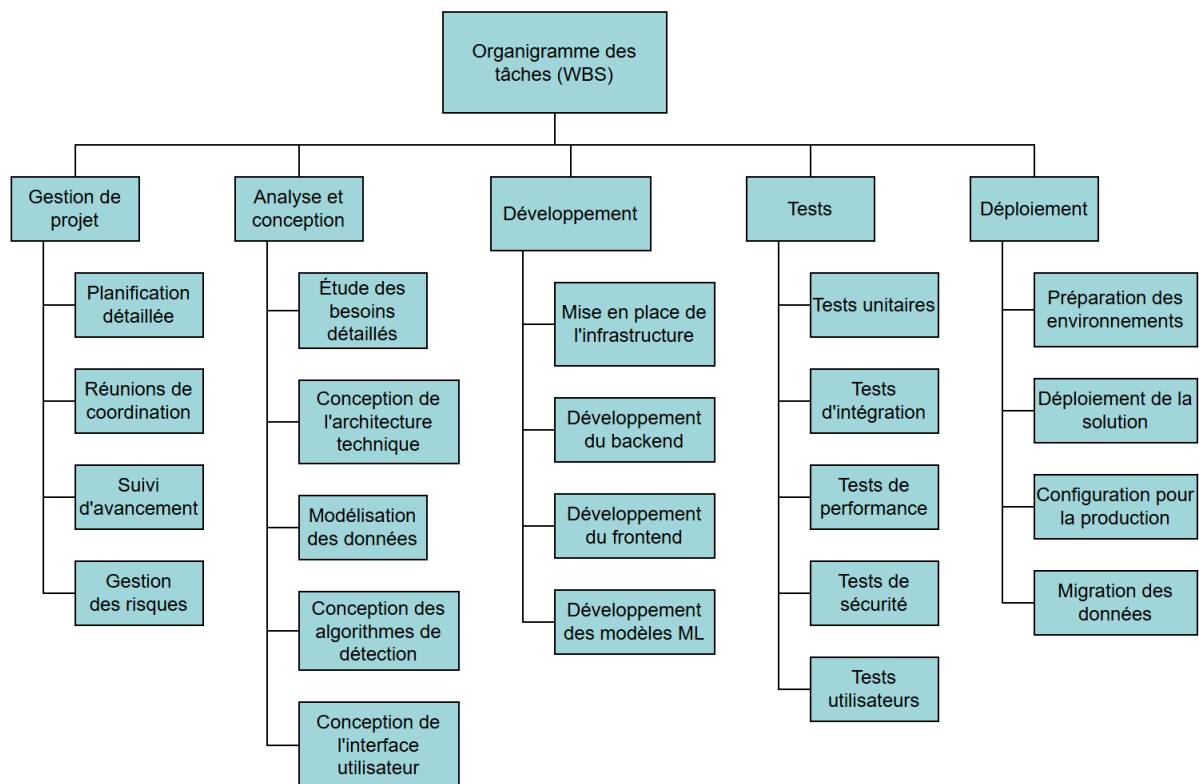


FIGURE 3.1 – Organigramme des Tâches (WBS)

3. Détail des Tâches Structurées

1. Gestion de projet
 - (a) Planification détaillée
 - (b) Réunions de coordination
 - (c) Suivi d'avancement
 - (d) Gestion des risques
 - (e) Clôture du projet
2. Analyse et conception
 - (a) Étude des besoins détaillés
 - (b) Conception de l'architecture technique
 - (c) Modélisation des données
 - (d) Conception des algorithmes de détection
 - (e) Conception de l'interface utilisateur
 - (f) Validation de la conception
3. Développement
 - (a) Mise en place de l'infrastructure
 - i. Configuration des serveurs
 - ii. Installation des outils Big Data
 - iii. Configuration des bases de données
 - (b) Développement du backend
 - i. API de connexion bancaire
 - ii. Moteur d'analyse des transactions
 - iii. Système de détection de fraude
 - iv. Moteur de règles personnalisables
 - (c) Développement du frontend
 - i. Interface de tableau de bord
 - ii. Écrans de configuration
 - iii. Système de notification
 - (d) Développement des modèles ML
 - i. Collecte et préparation des données
 - ii. Création des modèles prédictifs
 - iii. Entraînement des modèles
 - iv. Optimisation des performances

4. Tests

- (a) Tests unitaires
- (b) Tests d'intégration
- (c) Tests de performance
- (d) Tests de sécurité
- (e) Tests utilisateurs

5. Déploiement

- (a) Préparation des environnements
- (b) Déploiement de la solution
- (c) Configuration pour la production
- (d) Migration des données

6. Formation et support

- (a) Élaboration du matériel de formation
- (b) Formation des utilisateurs
- (c) Mise en place du support

4 Estimation de la durée

L'estimation de la durée du projet permet de planifier efficacement les ressources, les efforts et les échéances. Elle repose sur une analyse approfondie des différentes phases du projet et sur l'expérience acquise lors de projets similaires. Cette estimation prend également en compte les éventuels chevauchements entre certaines phases, comme la gestion de projet qui s'étend sur toute la durée.

Durées estimées par phase :

Phase	Durée estimée (semaines)
Gestion de projet	40 (tout au long du projet)
Analyse et conception	8
Développement	20
Tests	6
Déploiement	4
Formation et support	2

TABLE 4.1 – Durée estimée des phases du projet

Durée totale estimée du projet : 40 semaines (10 mois)

Troisième partie

Ordonnancement des tâches (Diagramme de Gantt)

1. Définition du diagramme de Gantt

Le **diagramme de Gantt** est un outil très utile pour la gestion de projet car il permet de visualiser de manière claire et concise l'état d'avancement des différentes tâches du projet. Il permet également de déterminer les dépendances entre les tâches, les durées estimées, les échéances, les ressources nécessaires et les contraintes potentielles.

2. Tableau d'Ordonnancement des Tâches

Tâche	Début	Fin	Durée (semaines)	Dépendances
1. Gestion de projet	S1	S40	40	-
2. Analyse et conception	S1	S8	8	-
3.1. Mise en place de l'infrastructure	S9	S11	3	2
3.2. Développement du backend	S12	S21	10	3.1
3.3. Développement du frontend	S14	S21	8	3.1
3.4. Développement des modèles ML	S12	S25	14	3.1
4.1. Tests unitaires	S15	S29	15	-
4.2-4.5. Tests d'intégration et autres	S26	S31	6	3.2, 3.3, 3.4
5. Déploiement	S32	S35	4	4
6. Formation et support	S36	S37	2	5
Clôture du projet	S38	S40	3	6

TABLE 4.2 – Ordonnancement des tâches

3. Diagramme de Gantt

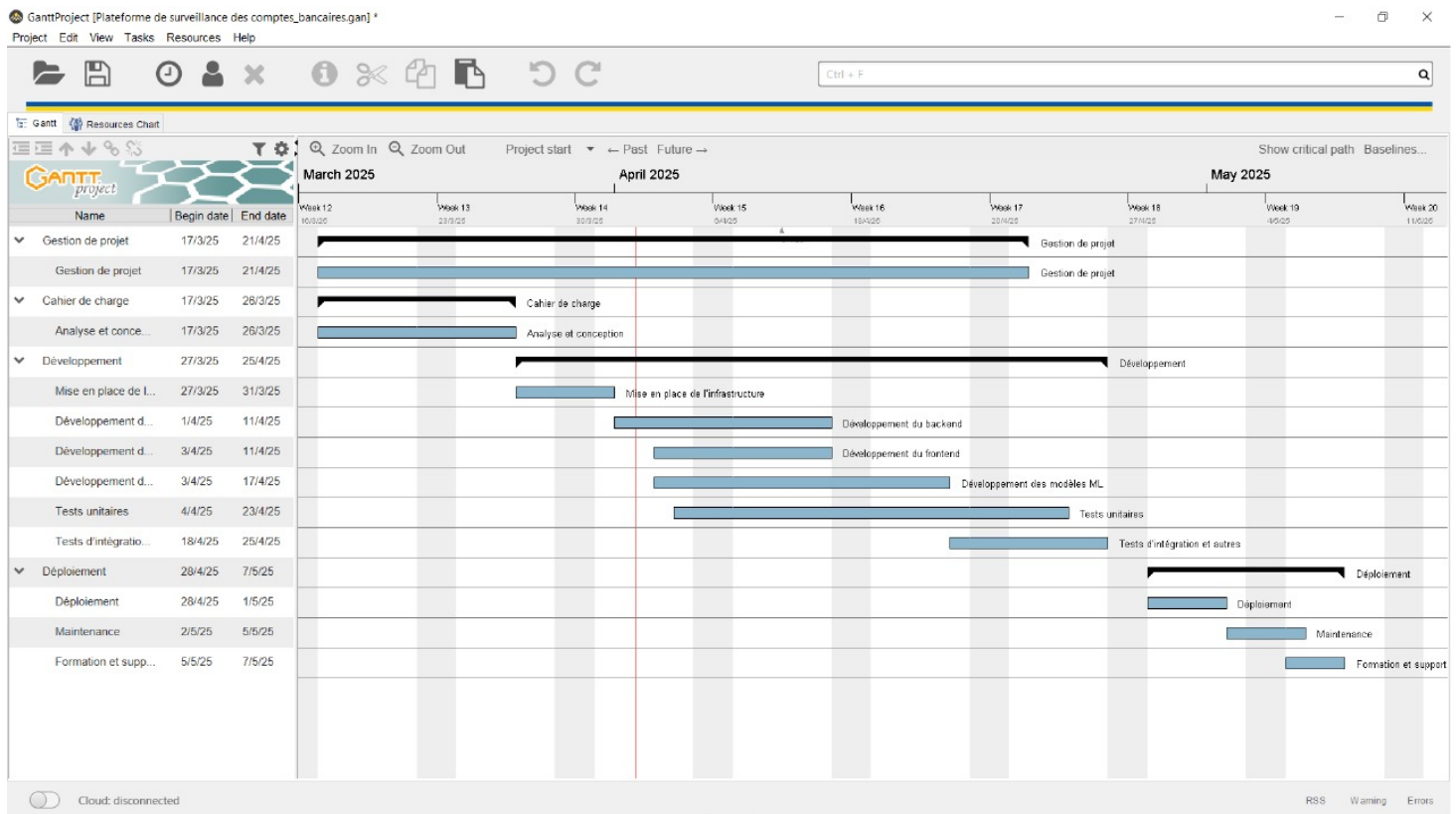


FIGURE 4.1 – Diagramme de Gantt du projet

Quatrième partie

Technique de PERT (Chemin critique)

1. Tableau des tâches avec durées et dépendances

Tâche	Description	Durée	Prédéc.	Tâches suivantes	Début tôt	Fin tôt	Début tard	Fin tard	Marge
A	Analyse et conception	8	-	B	S1	S8	S1	S8	0
B	Mise en place infrastructure	3	A	C, D, E	S9	S11	S9	S11	0
C	Développement backend	10	B	G	S12	S21	S12	S21	0
D	Développement frontend	8	B	G	S14	S21	S14	S21	0
E	Développement modèles ML	14	B	G	S12	S25	S12	S25	0
F	Tests unitaires	15	-	-	S15	S29	S15	S29	0
G	Tests d'intégration et autres	6	C,D,E	H	S26	S31	S26	S31	0
H	Déploiement	4	G	I	S32	S35	S32	S35	0
I	Formation et support	2	H	J	S36	S37	S36	S37	0
J	Clôture du projet	3	I	-	S38	S40	S38	S40	0

TABLE 4.3 – Tableau des tâches pour le diagramme PERT

Chemin critique : A → B → E → G → H → I → J (Total : 40 semaines)

2. Diagramme de PERT

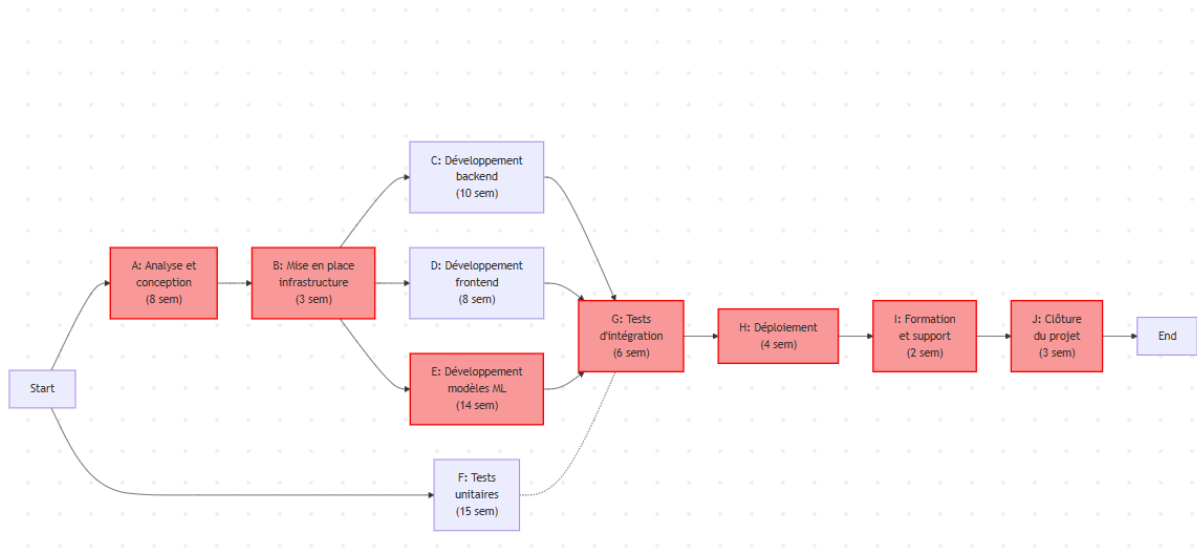


FIGURE 4.2 – Diagramme PERT du projet montrant le chemin critique

Cinquième partie

Management des risques et Matrice des Compétences

1. Matrice de gestion des risques

Une matrice de risque, également appelée matrice de probabilité-impact ou matrice de gestion des risques, est un outil utilisé pour évaluer et prioriser les risques dans un projet, une organisation ou une activité. Elle permet de visualiser et de classer les risques en fonction de leur probabilité d'occurrence et de leur impact sur les objectifs.

Risque	Prob.	Impact	Criticité	Stratégie de mitigation
Retard dans le développement des algorithmes ML	4	5	20	Affecter des ressources supplémentaires, commencer par des modèles simples puis raffiner
Problèmes de performance avec les données volumineuses	3	5	15	Tests de charge précoces, architecture évolutive, options de mise à l'échelle
Non-conformité réglementaire	2	5	10	Audit régulier par expert juridique, veille réglementaire constante
Faux positifs dans la détection de fraude	4	4	16	Phase d'apprentissage supervisé, mécanismes de feedback utilisateur
Problèmes d'intégration avec les systèmes bancaires	3	4	12	Prototypage précoce, API de test, établir des partenariats bancaires tôt
Failles de sécurité	2	5	10	Tests de pénétration réguliers, audits de sécurité externes
Indisponibilité des experts ML	3	4	12	Formation croisée de l'équipe, documentation détaillée, consultants externes
Dépassement budgétaire	3	3	9	Suivi financier rigoureux, jalons clairs, marge de contingence

TABLE 4.4 – Analyse et gestion des risques

2. Matrice des Compétences

2.1 Présentation de la matrice

Une matrice de compétences est un tableau structuré qui permet de visualiser les compétences requises pour un projet, un poste ou une activité spécifique.

Elle aide à évaluer les niveaux de maîtrise de vos collaborateurs sur leurs compétences et à identifier leurs besoins en formation.

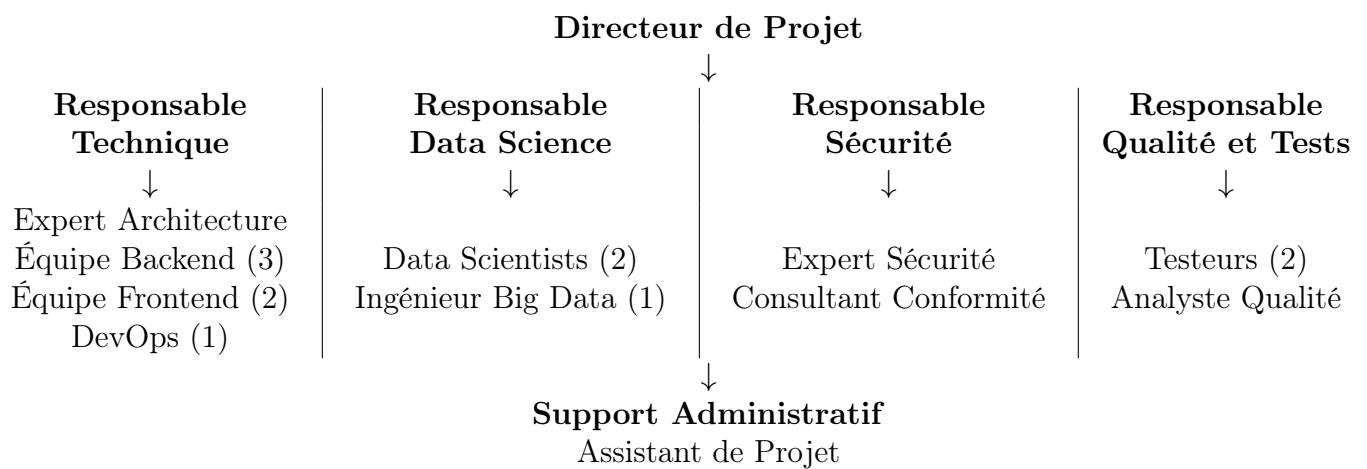
Compétence	Niveau (1-5)	Dispo. interne	Action
Gestion de projet	4	Oui	Formation avancée pour le chef de projet
Développement Big Data	5	Partielle	Recrutement d'un expert, formation interne
Machine Learning	5	Non	Recrutement de 2 data scientists
Sécurité informatique	5	Partielle	Consultant externe + formation interne
Développement backend	4	Oui	Équipe existante suffisante
Développement frontend	3	Oui	Équipe existante suffisante
Règlementations bancaires	4	Non	Consultant juridique spécialisé
Architecture système	4	Oui	Formation complémentaire
DevOps	4	Partielle	Recrutement d'un spécialiste
Tests et qualité	4	Oui	Formation spécifique aux tests de sécurité

TABLE 4.5 – Matrice des compétences requises

3. Organigramme fonctionnel (OBS)

Un organigramme des tâches du projet (OTP) ou structure de découpage du projet (SDP), également appelé *Work Breakdown Structure (WBS)* en anglais, est une décomposition hiérarchique des travaux nécessaires pour réaliser les objectifs d'un projet.

Cet outil permet d'organiser efficacement le projet en :



Sixième partie

Estimation des coûts internes et externes

1. Estimation des coûts internes

L'estimation des coûts internes est une étape clé dans la planification d'un projet. Elle permet d'évaluer les ressources nécessaires à la réalisation du projet, qu'elles soient humaines, matérielles ou logicielles, afin de garantir une gestion budgétaire maîtrisée.

Le tableau ci-dessous présente une estimation détaillée des coûts liés aux ressources humaines, matérielles et logicielles mobilisées pour le projet.

Ressource	Coût (€/jour)	Nb jours	Coût total (€)
Directeur de Projet	800	200	160 000
Responsable Technique	700	200	140 000
Expert Architecture	650	100	65 000
Développeurs Backend (3)	550	600	330 000
Développeurs Frontend (2)	550	400	220 000
DevOps	600	150	90 000
Data Scientists (2)	700	400	280 000
Ingénieur Big Data	650	150	97 500
Expert Sécurité	750	100	75 000
Testeurs (2)	500	300	150 000
Analyste Qualité	550	150	82 500
Assistant de Projet	350	200	70 000
Total ressources humaines			1 760 000
Matériel informatique			50 000
Licences logicielles			80 000
Formation interne			30 000
Total coûts internes			1 920 000

TABLE 4.6 – Estimation des coûts internes

2. Estimation des coûts externes

L'estimation des coûts externes permet d'identifier les dépenses liées aux prestations, services ou ressources qui ne sont pas disponibles en interne et doivent être externalisées. Ces coûts incluent les consultants, les services spécialisés, les infrastructures hébergées à l'extérieur (comme le cloud), ainsi que les licences spécifiques nécessaires à la réussite du projet.

Le tableau ci-dessous détaille les ressources externes mobilisées, leur coût unitaire, la quantité estimée, ainsi que le coût total associé.

Ressource externe	Coût unitaire (€)	Quantité	Coût total (€)
Consultant conformité réglementaire	1 000/jour	50 jours	50 000
Consultant sécurité externe	1 200/jour	30 jours	36 000
Infrastructure cloud (par mois)	15 000	10 mois	150 000
Services d'audit de sécurité	40 000	1	40 000
Licences API bancaires	25 000	1	25 000
Services juridiques	800/jour	20 jours	16 000
Formation externe	2 500/personne	8 personnes	20 000
Total coûts externes			337 000

TABLE 4.7 – Estimation des coûts externes

Budget total du projet : 2 257 000 €

Conclusion Générale

En résumé, ce projet de plateforme de surveillance bancaire intelligente illustre parfaitement l'alliance stratégique entre Big Data, intelligence artificielle et architecture microservices pour répondre aux défis contemporains de la fraude financière. Grâce à une détection efficace des fraudes en temps réel, une capacité de traitement élevée et une conformité stricte aux régulations, la solution proposée combine innovation technologique, viabilité économique et responsabilité éthique. Forte d'apports techniques majeurs tels que l'intégration de flux hétérogènes, l'apprentissage adaptatif et la visualisation avancée, la plateforme offre une réponse robuste et évolutive aux besoins des institutions financières. Elle ouvre la voie à de nouvelles perspectives d'extension vers les cryptomonnaies et les fraudes génératives, affirmant ainsi son rôle clé dans la sécurisation des actifs numériques à l'ère de l'économie 4.0.