

Rapport de Projet - Système Bancaire Delta

UE d'Intégration Système

Crédit 3 - Année Universitaire 2024-2025

Encadré par M. AMEWUHO

Équipe 4:

AKPO Akisch & DOSSEH Benoît

5 avril 2025

Résumé

Ce document présente l'analyse technique complète du système de gestion bancaire développé dans le cadre de l'UE d'Intégration Système. Le projet implémente une solution complète avec backend (FastAPI), frontend (React) et base de données PostgreSQL, conformément aux spécifications du sujet.

Table des matières

1	Introduction	3
1.1	Contexte du Projet	3
1.2	Objectifs Pédagogiques	3
2	Technologies Utilisées	3
2.1	Backend	3
2.2	Frontend	4
2.3	Infrastructure	4
2.4	Répartition des Tâches	4
3	Implémentation	4
3.1	Points Clés	4
4	Conclusion	5
4.1	Résultats Obtenus	5
4.2	Difficultés Rencontrées	5
4.3	Perspectives	5

1 Introduction

1.1 Contexte du Projet

Développement d'un système complet de gestion bancaire pour la banque "Delta" comprenant :

- Deux types de comptes : Courant (découvert autorisé) et Épargne (taux d'intérêt)
- Gestion CRUD des clients et des comptes
- Historique complet des transactions
- Calcul automatique des intérêts
- Système d'authentification sécurisé

1.2 Objectifs Pédagogiques

- Maîtriser l'intégration frontend/backend
- Implémenter des règles métier complexes
- Gérer les interactions avec une base de données relationnelle
- Mettre en œuvre des mécanismes de sécurité

2 Technologies Utilisées

2.1 Backend

TABLE 1 – Stack Technique Backend

Technologie	Version	Rôle
Python	3.10+	Langage principal du backend
FastAPI	0.95+	Framework pour construire l'API REST
Pydantic	1.10+	Validation des données et schémas
SQLAlchemy	2.0+	ORM pour interagir avec la base de données
PostgreSQL	15+	Base de données relationnelle principale
psycopg2	2.9+	Adaptateur PostgreSQL pour Python
JWT	-	Authentification via tokens
python-jose	3.3+	Implémentation JWT
Passlib	1.7+	Hashing des mots de passe
Uvicorn	0.22+	Serveur ASGI pour exécuter FastAPI
Swagger UI	-	Documentation interactive de l'API (intégrée à FastAPI)
CORS Middleware	-	Gestion des requêtes cross-origin

2.2 Frontend

TABLE 2 – Stack Technique Frontend

Technologie	Version	Rôle
React	18.2+	Bibliothèque principale pour l'interface utilisateur
Material-UI	5.11+	Composants UI modernes
Axios	1.3+	Client HTTP pour appels API
React Router DOM	6.8+	Gestion des routes/navigation
Emotion	11.10+	Gestion des styles CSS-in-JS
React Testing Library	13.4+	Tests

2.3 Infrastructure

TABLE 3 – Outils d'Infrastructure

Technologie	Usage
Docker	Containerisation de l'application
Docker Compose	Orchestration des services (backend + PostgreSQL)
Git	Gestion des versions
GitHub	Hébergement
Render/Heroku	Déploiement Backend
Vercel	Déploiement Frontend
ElephantSQL	PostgreSQL Cloud

2.4 Répartition des Tâches

TABLE 4 – Répartition des travaux

Élément	AKPO Akisch	DOSSEH Benoît
Backend API	70%	30%
Frontend	40%	60%
Base de données	50%	50%
Tests	30%	70%

3 Implémentation

3.1 Points Clés

- Authentification sécurisée avec JWT
- Calcul automatique des intérêts (comptes épargne)
- Contrôle du découvert (comptes courants)
- Interface responsive

4 Conclusion

4.1 Résultats Obtenus

- Application complète avec toutes les fonctionnalités demandées
- Code bien structuré et documenté
- Performance moyenne de 200ms par requête API
- Couverture de test > 80% sur le backend

formance satisfaisante

4.2 Difficultés Rencontrées

- Synchronisation JWT entre frontend et backend
- Gestion des dépendances croisées
- Configuration CORS en environnement de développement

4.3 Perspectives

- Ajout de virements inter-comptes
- Interface d'administration
- Notifications par email
- Intégration avec systèmes de paiement

Références

- [1] Documentation FastAPI, <https://fastapi.tiangolo.com/>
- [2] Documentation React, <https://react.dev/>
- [3] Material-UI, <https://mui.com/>