

CAHIER CHARGE DU PROJET

Contexte du projet :

Le projet est pour but de développer une application qui permet de gérer les transactions du paiement facture et livraison. Le projet se divise en trois parties, la partie Agent/BackOffice qui permet la souscription des agents et clients pour bénéficier du service de paiement facture, la deuxième partie est celle de l'espace de client, et la dernière partie est celle du web service de CMI qui se représente dans ce projet comme un Core Banking qui permet d'autoriser les paiement client.

Composant du projet :

1. **Application BackOffice** (Spring Boot + Angular) :
 - Gestion des agents et clients.
 - Création et gestion des comptes utilisateurs avec un formulaire sécurisé.
 - Système d'authentification et de gestion des rôles (Admin, Agent, Client).
 - Création et gestion de cartes virtuelles pour les clients.
2. **Application Agent** (Spring Boot + Angular) :
 - Interface pour les agents permettant d'ajouter de nouveaux clients.
 - Gestion des souscriptions aux services de portefeuille.
 - Consultation des comptes et gestion des cartes virtuelles pour les clients.
3. **Application Web Client** (Angular) :
 - **Authentification sécurisée** via JWT.
 - **Consultation des soldes** des portefeuilles et des cartes virtuelles.
 - **Paiement entre utilisateurs** (transfert d'argent entre deux comptes clients).
 - **Module de paiement de factures** similaire à ENSA PAY (factures électricité, eau, télécoms).
 - **Achats via le portefeuille** chez des marchands partenaires.

Informations personnelles

Nom *	Prénom *
<input type="text"/>	<input type="text"/>
Pièce d'identité *	N° pièce d'identité *
<input type="text" value="C.I.N"/>	<input type="text"/>
Date de naissance	Adresse *
<input type="text"/>	<input type="text"/>
Email *	Confirmation Email
<input type="text"/>	<input type="text"/>
Numéro de téléphone *	N° d'immatriculation au registre de commerce *
<input type="text"/>	<input type="text"/>
N° de patente *	
<input type="text"/>	

Pièces jointes

Description	Fichier
<input type="text" value="Description"/>	<input type="button" value="Parcourir..."/> <input type="button" value="Auc...né."/> <input type="button" value="✚"/>

✓ Enregistrer

✓ Annuler

Application Agent :

Ouverture d'un compte de paiement

Sélectionner un produit

Hssab 1 - Plafond : 200 DH

Informations client

Nom

Prénom

Téléphone

Email

✓ Suivant


Application Web Client :

Paielement de facture


LISTE DES CRÉANCIERS

HISTORIQUE


Toutes les catégories




IAM RECHARGES
» TÉLÉPHONIE ET INTERNET SIM




IAM FACTURES
» PRODUIT INTERNET SIM
» PRODUIT FIXE SIM
» PRODUIT MOBILE SIM



REDAL
» FACTURES REDAL




AMENDIS TANGER
» FACTURES AMENDIS TANGER



LYDEC

AMENDIS TETOUAN

Effectuer un paiement



ALCS
association
de lutte
contre
le sida

PAIEMENT DE
FACTURES ET RECHARGES

Nom et Prénom du donateur

Montant du don (DH)


✕ Annuler

✓ Valider

Commission : Commission indisponible

Paielement de facture

Paielement par référence



Créancier

ALCS

Créance

SIDACTION

Donateur

ANAS

Sélectionnez pour payer

Don Sidaction10.00 DH


Total0.00 DH

Valider

Commission : Commission indisponible

Paielement de facture

Paielement par référence



Créancier

ALCS

Date de création

23-12-2020

Créance

SIDACTION

Donateur

ANAS

Récapitulatif des informations saisies

RÉFÉRENCE	DESCRIPTION	PRIX TTC
100001192072	Don Sidaction	10 DH
Total		10 DH

Abandonner la transaction

Confirmer et Signer

Règles de gestion :

- Le solde client est récupéré par web service depuis CMI.
- Le module n'est pas accessible si le solde client est différent à 0.
- Le formulaire est récupéré et généré dynamiquement depuis le web service
- Les impayés « Frais » et « Pénalité » sont sélectionnés par défaut.
- La signature de la demande se fait via un web service qui permet de vérifier le solde client et confirmer le paiement.

Fonctionnalités principales :

1. **Gestion des comptes :**
 - Création de comptes avec la validation d'un OTP.
 - Gestion des soldes en temps réel via un service bancaire tiers.
 - Liaison de comptes bancaires pour les recharges de portefeuille.
2. **Transactions et paiements :**
 - Transferts d'argent entre utilisateurs de l'application.
 - Paiement de factures via web service.
 - Possibilité d'ajouter des cartes virtuelles pour sécuriser les achats en ligne.
 - Paiement récurrent (par exemple, pour des abonnements).
3. **Gestion des cartes virtuelles :**
 - Les utilisateurs peuvent créer des cartes virtuelles temporaires pour sécuriser les paiements en ligne.
 - Les cartes peuvent être utilisées chez des marchands en ligne.
4. **Batch de réconciliation :**
 - Un système de batch vérifie quotidiennement les écarts entre les transactions effectuées et les confirmations des prestataires de paiement, avec possibilité de corrections automatiques.
5. **Notifications en temps réel :**
 - Les clients reçoivent des notifications push/SMS pour chaque transaction ou mise à jour de solde.

Technologies utilisées :

- **Back-end** : Spring Boot, Spring Data JPA, Spring Security, Feign pour les appels aux API tierces.
- **Front-end** : Angular avec Angular Material pour l'interface utilisateur.
- **Base de données** : MySQL ou PostgreSQL pour la gestion des utilisateurs, transactions, et portefeuilles.
- **Web services** :
 - Web services RESTful pour la communication entre les microservices.
 - Intégration de SOAP pour interagir avec les systèmes de paiement tiers.
- **Sécurité** : Utilisation de JWT pour sécuriser les sessions utilisateur.

Déploiement : Conteneurisation avec Docker, orchestration avec Kubernetes

Étapes de réalisation :

1. Conception :

- Modélisation des entités clés (Utilisateurs, Transactions, Cartes virtuelles).
- Création des diagrammes UML (cas d'utilisation, diagrammes de classes et de séquence).

2. Développement back-end :

- Création des microservices pour la gestion des utilisateurs, transactions, cartes virtuelles.
- Intégration des web services de paiement tiers.
- Implémentation des batchs pour la réconciliation des paiements.

3. Développement front-end :

- Création d'une interface utilisateur moderne avec Angular pour gérer les paiements et les transactions.
- Intégration des web services pour consulter les soldes et effectuer des paiements.

4. Tests et déploiement :

- Tests unitaires et d'intégration.
- Déploiement de l'application via Docker.

Fonctionnalité

Paielements multi-devises

- **Description** : Permettre à l'utilisateur d'effectuer des paiements et de gérer son portefeuille en différentes devises.
- **Fonctionnalités** :
 - **Portefeuille multi-devises** : Permet aux utilisateurs de gérer des comptes en EUR, USD, etc.
 - **Conversion de devises** en temps réel avec des taux de change dynamiques.
 - **Paiement international** via des API externes pour gérer les transactions en devises étrangères.

Paielements récurrents automatisés

- **Description** : Permettre aux utilisateurs de configurer des paiements automatiques pour des services comme les abonnements (factures d'électricité, téléphonie, etc.).
- **Fonctionnalités** :
 - **Planification de paiements récurrents** (mensuels, hebdomadaires).
 - **Gestion et annulation** des paiements récurrents.
 - **Notification avant prélèvement** pour informer l'utilisateur de la transaction programmée.

Gestion des dépenses et budget personnel

- **Description** : Intégrer un outil d'analyse des dépenses pour permettre aux utilisateurs de mieux gérer leurs finances personnelles.
- **Fonctionnalités** :
 - **Catégorisation des dépenses** : Les dépenses sont automatiquement classées (loyer, alimentation, loisirs, etc.).
 - **Graphiques d'analyse** : Statistiques et graphiques pour suivre l'évolution des dépenses mensuelles ou annuelles.
 - **Définition de budgets** : Les utilisateurs peuvent définir des limites budgétaires et recevoir des alertes en cas de dépassement.

Portefeuille pour les cryptomonnaies

- **Description** : Ajouter la gestion des cryptomonnaies (comme le Bitcoin, Ethereum) dans l'application pour permettre aux utilisateurs d'échanger ou d'acheter des cryptos.
- **Fonctionnalités** :
 - **Acheter/Vendre des cryptomonnaies** à partir du portefeuille.
 - **Transfert entre cryptos et argent classique** (via une API d'échange de cryptomonnaies).
 - **Suivi des transactions de cryptos** avec un historique et un graphique du taux de change