

Cahier des Charges: **Microservice de Paiements de Masse**

1. Introduction

1.1 Objectif du Document

Ce document présente les spécifications techniques et fonctionnelles pour le développement d'un microservice de paiements de masse destiné à une application mobile bancaire.

1.2 Portée du Projet

Le microservice permettra aux utilisateurs d'effectuer des transferts d'argent simultanés vers plusieurs destinataires en une seule opération, y compris vers des comptes situés dans différentes banques ou portefeuilles électroniques. Il inclut également la gestion des groupes de destinataires pour faciliter les paiements récurrents, ainsi que la possibilité d'uploader un fichier CSV pour ajouter des destinataires à un groupe.

1.3 Définitions

- **Paiement de masse** : Opération permettant de transférer de l'argent vers plusieurs comptes simultanément.
- **Transaction** : Mouvement d'argent individuel faisant partie d'une opération de paiement de masse.
- **Initiateur** : L'utilisateur qui initie le paiement de masse.
- **Destinataire** : Un utilisateur qui reçoit des fonds dans le cadre d'un paiement de masse.
- **Transaction inter-banques** : Transfert entre des comptes appartenant à des établissements bancaires différents.
- **Groupe de Destinataires** : Une collection de destinataires enregistrés pour faciliter les paiements récurrents.
- **CSV Upload** : Fonctionnalité permettant d'ajouter des destinataires à un groupe via un fichier CSV.

2. Description Fonctionnelle

2.1 Fonctionnalités Principales

1. Créer un paiement de masse vers plusieurs destinataires, y compris vers des comptes externes.
2. Vérifier l'éligibilité de chaque transaction individuelle.

3. Exécuter les transactions individuelles indépendamment les unes des autres.
4. Conserver l'historique des paiements de masse et des transactions individuelles.
5. Gérer les routages des transactions vers différentes banques partenaires.
6. Gestion des Groupes de Destinataires :
 - Créer, mettre à jour et supprimer des groupes de destinataires.
 - Ajouter ou supprimer des destinataires d'un groupe.
 - Valider les destinataires avant de les ajouter à un groupe.
 - Effectuer des paiements de masse à partir d'un groupe de destinataires.
 - Upload de fichiers CSV pour ajouter des destinataires à un groupe.

2.2 Règles Métier

1. L'initiateur doit disposer d'un solde suffisant pour couvrir la totalité des paiements et frais.
2. Les comptes destinataires doivent être actifs et non bloqués.
3. Chaque transaction est traitée indépendamment - l'échec d'une transaction n'affecte pas les autres.
4. Un message d'erreur précis doit être fourni pour chaque transaction échouée.
5. Le système débitera le compte de l'initiateur uniquement pour les transactions réussies.
6. Les délais de traitement peuvent varier selon la banque destinataire.
7. Règles pour les Groupes de Destinataires :
 - Un destinataire ne peut être ajouté à un groupe que s'il est valide (vérification du numéro de téléphone et du code banque).
 - Les destinataires peuvent être associés à un montant par défaut et un motif de paiement.
 - Les paiements de masse à partir d'un groupe doivent respecter les mêmes règles que les paiements de masse individuels.
 - CSV Upload :
 - Le fichier CSV doit contenir les colonnes suivantes : phone_number, bank_code, full_name, default_amount, motive.
 - Les données du CSV doivent correspondre à des enregistrements existants dans le système (utilisateur et compte actif).
 - Les lignes invalides seront ignorées et les erreurs seront retournées dans la réponse.

3. Architecture Technique

3.1 Modèle de Données

3.1.1 Modèles Existants

User:

- phone_number (unique)
- first name

- last_name

Account:

- user (relation)
- account_number (unique)
- balance
- is_active
- is_blocked
- bank_code (identifiant de la banque/portefeuille)

Transaction:

- transaction_type (transfert, withdrawal, deposit, payment)
- status (success, failure, pending)
- amount
- source_account (relation)
- destination_account (relation)
- fee_amount

BankProvider:

- bank_code (unique)
- name
- is_active
- api_endpoint

3.1.2 Nouveaux Modèles (Proposés)

MassPayment:

- initiator_account (relation vers Account)
- total_amount
- fee_amount
- status (pending, processing, completed, failed, partially_completed)
- created_at
- updated_at
- description
- reference_code (unique)
- success_count
- failure_count
- pending_count

PaymentTemplate:

- name
- owner (relation vers User)
- is_active
- created_at
- updated_at

MassPaymentItem:

- mass_payment (relation vers MassPayment)
- destination_phone_number
- destination_account (relation vers Account, optionnel)
- destination_bank_code

- amount
- status (pending, success, failed, processing)
- transaction (relation vers Transaction, optionnel)
- fee_amount

TemplateRecipient:

- template (relation vers PaymentTemplate)
- phone_number
- bank_code
- name
- default_amount (optionnel)

RecipientGroup:

- name
- owner (relation vers User)
- is_active
- created_at
- updated_at
- status (pending, processing, completed, failed, partially_completed)

GroupRecipient:

- group (relation vers RecipientGroup)
- phone_number
- bank_code
- full_name
- default_amount
- motive
- status (pending, validated, failed)

3.2 Endpoints API

3.2.1 Création d'un Paiement de Masse

POST /api/mass-payments

Requête:

```
{
  "initiator_account_number": "string",
  "recipients": [
    {
      "phone_number": "string",
      "bank_code": "string",
      "amount": decimal
    }
  ],
  "description": "string",
  "reference": "string" (optionnel)
}
```

Réponse:

```
{
  "mass_payment_id": "string",
  "status": "string",
  "total_amount": decimal,
  "fee_amount": decimal,
  "created_at": "datetime",
  "recipients_count": integer,
  "external_recipients_count": integer,
  "estimated_completion_time": "datetime"
}
```

3.3.2 Gestion des Groupes de Destinataires

- step1: Create a New Recipient Group

POST /api/recipient-groups

Requête:

```
{
  "name": "string"
}
```

Réponse:

```
{
  "id": "string",
  "name": "string",
  "is_active": boolean,
  "created_at": "datetime"
  "updated_at": "datetime"
}
```

- step2: Validate a Recipient

POST /api/recipient-groups/validate_recipient/

Requête:

```
{
  "phone_number": "string",
  "bank_code": "string",
}
```

Réponse:

```
{
  "exists": boolean,
  "full_name": "string"
  "phone_number": "string"
  "account_number": "string"
}
```

- Step 3: Add a Recipient to the Group

```
POST /api/recipient-groups/{group_id}/add_recipient/
```

Requête:

```
{
  "phone_number": "string",
  "bank_code": "string",
  "default_amount": decimal,
  "motive": "string"
}
```

Réponse:

```
{
  "success": boolean,
  "id": "string",
  "full_name": "string"
  "phone_number": "string"
  "bank_code": "string"
  "default_amount": decimal
  "motive": "string"
  "status": "string"
}
```

- Step 4: Create a Mass Payment from the Group

```
POST /api/recipient-groups/{group_id}/create-mass-payment
```

Requête:

```
{
  "initiator_account_number": "string",
  "description": "string"
}
```

Réponse:

```
{
  "mass_payment_id": "string",
  "reference_code": "string",
  "status": "string",
  "total_amount": decimal,
  "fee_amount": decimal,
  "created_at": "datetime",
  "recipients_count": integer,
  "external_recipients_count": integer,
  "estimated_completion_time": "datetime",
  "recipients": []
}
```

- Upload de Destinataires via CSV :

```
POST /api/recipient-groups/{group_id}/create-mass-payment
```

Requête:

```
{
  "initiator_account_number": "string",
  "description": "string"
}
```

Réponse:

```
{
  "mass_payment_id": "string",
  "reference_code": "string",
  "status": "string",
  "total_amount": decimal,
  "fee_amount": decimal,
  "created_at": "datetime",
  "recipients_count": integer,
  "external_recipients_count": integer,
  "estimated_completion_time": "datetime",
  "recipients": []
}
```

3.2.2 Récupération des Détails d'un Paiement de Masse

```
GET /api/mass-payments/{mass_payment_id}
```

Réponse:

```
{
  "mass_payment_id": "string",
  "initiator_account_number": "string",
  "status": "string",
  "total_amount": decimal,
  "fee_amount": decimal,
  "success_count": integer,
  "failure_count": integer,
  "pending_count": integer,
  "created_at": "datetime",
  "updated_at": "datetime",
  "description": "string",
  "reference": "string",
  "contains_external_transfers": boolean,
  "items": [
    {
      "destination_phone_number": "string",
      "destination_account_number": "string",
      "destination_bank_code": "string",
      "bank_name": "string",
      "amount": decimal,

```

```

    "fee_amount": decimal,
    "status": "string",
    "estimated_completion_time": "datetime",
    "external_reference": "string",
    "failure_reason": "string"
  }
]
}

```

3.2.3 Liste des Paiements de Masse d'un Utilisateur

```
GET /api/accounts/{account_number}/mass-payments
```

3.2.4 Gestion des Templates de Paiement

- POST /api/payment-templates
- GET /api/payment-templates
- GET /api/payment-templates/{template_id}
- PUT /api/payment-templates/{template_id}
- DELETE /api/payment-templates/{template_id}

3.3 Processus de Traitement

3.3.1 Modèles Existants

Avant de traiter un paiement de masse, le système doit effectuer les vérifications suivantes:

1. Vérifier que le compte initiateur existe et est actif
2. Vérifier que le solde du compte initiateur est suffisant pour le montant total et les frais estimés
3. Pour chaque transaction individuelle:
 - Vérifier l'existence du numéro de téléphone du destinataire
 - Vérifier que le compte destinataire est actif et non bloqué
 - Vérifier que la transaction respecte les limites configurées
 - Pour les transferts inter-banques, vérifier que la banque destinataire est Supportée

3.3.1 Traitement des Transactions

Le traitement du paiement de masse doit suivre ces étapes

1. Créer l'enregistrement MassPayment principal avec le statut "processing"
2. Créer un enregistrement MassPaymentItem pour chaque destinataire
3. Pour chaque transaction :
 - Vérifier la disponibilité des fonds au moment de l'exécution
 - Si suffisant, débiter le compte de l'initiateur et créer la transaction
 - Marquer la transaction comme "success", "failure" ou "pending" selon le résultat
 - Mettre à jour les compteurs success_count, failure_count et pending_count

4. Pour les transactions internes, le traitement est généralement immédiat :
 5. Mettre à jour le statut global du **MassPayment** :
 - "completed" : Si toutes les transactions ont réussi.
 - "failed" : Si toutes les transactions ont échoué.
 - "partially_completed" : Si certaines transactions ont réussi et d'autres ont échoué.
 - "processing" : Tant qu'il reste des transactions en attente.
-

3.2.3 Traitement des Groupes de Destinataires

1. Création d'un Groupe de Destinataires :
 - L'utilisateur crée un groupe en spécifiant un nom.
 - Le groupe est initialement en statut "pending".
2. Ajout de Destinataires :
 - L'utilisateur ajoute des destinataires un par un.
 - Chaque destinataire est validé avant d'être ajouté au groupe.
 - Si la validation échoue, le destinataire n'est pas ajouté.
3. Validation des Destinataires :
 - Le système vérifie que le numéro de téléphone et le code banque du destinataire sont valides.
 - Si valide, le destinataire est ajouté au groupe avec le statut "validated".
 - Si invalide, le destinataire est marqué comme "failed" avec une raison d'échec.
4. Paiement de Masse à partir d'un Groupe :
 - L'utilisateur sélectionne un groupe et initie un paiement de masse.
 - Le système vérifie que tous les destinataires du groupe sont valides.
 - Le paiement de masse est traité comme un paiement de masse standard,

3.3.2 Traitement des Transactions

1. Créer l'enregistrement **MassPayment** principal avec le statut "processing".
 2. Créer un enregistrement **MassPaymentItem** pour chaque destinataire.
 3. Pour chaque transaction :
 - Vérifier la disponibilité des fonds au moment de l'exécution.
 - Si suffisant, débiter le compte de l'initiateur et créer la transaction.
 - Mettre à jour les compteurs `success_count`, `failure_count` et `pending_count`.
-

4. Conclusion

Le microservice de paiements de masse, avec l'ajout de la fonctionnalité d'upload de fichiers CSV, offre une solution robuste et flexible pour gérer les transferts d'argent en masse. La validation des destinataires et la gestion des erreurs garantissent une expérience utilisateur fluide et sécurisée.