

# Rapport

# Monnaie Virtuelle

Alexandre Cazottes  
Arman Hayrapetyan  
Bena Hugo  
Tuline Leveque  
Eloi Reignier

Semaine 13 25/03/2025

<b>Périmètre du projet</b> .....	<b>2</b>
<b>Existant</b> .....	<b>4</b>
<b>Acteurs</b> .....	<b>5</b>
<b>Fonctionnalités principales</b> .....	<b>6</b>
<b>Fonctionnalités secondaires</b> .....	<b>8</b>
<b>Fonctionnalités étendues</b> .....	<b>8</b>
<b>Diagrammes</b> .....	<b>9</b>
<b>Maquettes</b> .....	<b>13</b>
<b>Base de données</b> .....	<b>16</b>

Lien vers le CMS : <https://great-panini.163-5-143-4.plesk.page/>

# Périmètre du projet

Le but du projet est de créer un système de paiement sécurisé basé sur les principes de la blockchain, pour tout type d'événement tel que les festivals, marchés artisanaux, campus, etc.

Les utilisateurs (client ou commerçant) peuvent :

- Effectuer des paiements
- Consulter leur solde et historique
- Recevoir des paiements
- Emettre une facture

Les administrateurs peuvent :

- Créer un compte
- Supprimer un compte
- Voir les solde et historique d'un compte
- Effectuer des paiements (afin de mettre de l'argent sur un nouveau compte)
- Bloquer les transactions de la blockchain

La monnaie virtuelle doit répondre à la norme NF 525. Pour répondre à cette norme, le système doit répondre à certain critère :

- L'inaltérabilité et la sécurisation des données d'origine : Le système doit garantir que les données ne peuvent être modifiées.
- La conservation des données d'origine : Le système doit conserver les données de vente et d'encaissement, pour cela, le système doit faire des clôtures périodiques.

La monnaie électronique utilisée dans le cadre du Service est dénommée « Po ». Elle est libellée pour la valeur nominale que les fonds reçus en euros, selon la conversion suivante : 1 € = 100 Po.

Le compte "Nexus" permet à l'Utilisateur d'effectuer des paiements et de gérer ses "Po" au sein du système. L'usage du Compte "Nexus" est strictement personnel et doit se conformer aux présentes conditions d'utilisation.

L'Institution se réserve le droit, à tout moment et sans justification, de : clôturer le Compte "Nexus" et résilier l'accès au Service en cas de doute sur l'authenticité des documents d'identification, de soupçon de blanchiment ou de financement du terrorisme ; bloquer le Compte "Nexus" en cas de doute sur une utilisation anormale

des fonds et d'exiger des justificatifs complémentaires. Les opérations de rechargement suivies d'une ou plusieurs demandes de remboursement rapides peuvent être considérées comme suspectes ou frauduleuses. Le Service "Nexus" est fourni à titre gratuit, hors frais éventuels de communication ou de connexion facturés directement par les opérateurs téléphoniques ou fournisseurs d'accès à Internet.

Toute somme convertie en "Po" peut être échangée pour 1 "Po" = 0.01 €.



L'image de notre monnaie virtuelle, les "PO"

## Existant

Ce projet s'inspire de l'application et du système d'Izly, qui est utilisée dans les campus universitaires. Cette application permet de payer via son téléphone avec une monnaie virtuelle. Le rechargement du compte Izly se fait par carte bancaire. L'utilisateur a le choix entre plusieurs montants, pour recharger son compte : 5€, 10€, 20€, 30€ et 50€.

Ce système présente l'avantage d'être simple d'utilisation et largement adopté dans le milieu universitaire. Cependant, il dépend entièrement d'une connexion Internet et ne permet pas une utilisation optimale dans des environnements temporaires ou isolés, tels que les festivals, marchés artisanaux ou événements en plein air.

Dans notre cas, le compte pourra être alimenté de différentes manières :

- Carte bancaire sur le site et l'application.
- En liquide. Cela peut se faire à l'entrée du festival, où l'on propose des cartes, avec une somme déjà inscrite dessus. Il suffit simplement de relier cette carte au compte de l'utilisateur.

De plus, notre solution se distingue par son fonctionnement hors-ligne, rendu possible grâce à un réseau Wi-Fi local hébergé sur un Raspberry Pi, qui assure le stockage des données et la continuité du service en zone sans couverture réseau.

Ainsi, bien que notre projet s'inspire du fonctionnement d'Izly, il propose une solution plus adaptée aux événements temporaires, intégrant une gestion locale des transactions et un système sécurisé basé sur les principes de la blockchain.

# Acteurs

Admin :

- Créer un compte
- Supprimer un compte
- Voir les solde et historique d'un compte
- Effectuer des paiements (afin de mettre de l'argent sur un nouveau compte)
- Bloquer les transactions de la blockchain

Utilisateur (client et commerçant) :

- Effectuer des paiements
- Consulter leur solde et historique des transactions
- Recevoir des paiements
- Emettre une facture
- Demander une alimentation/rechargement du compte

# Fonctionnalités principales

Les utilisateurs :

- L'admin crée un compte utilisateur, alimente les comptes, vérifie la blockchain (table de transactions), et a la possibilité de bloquer toutes les transactions si une transaction semble suspecte ou si la table des transactions est corrompue. Si cela arrive, l'admin a la possibilité de revenir sur une sauvegarde de la table des transactions.
- L'utilisateur peut créer un compte, demander une alimentation, créer une facture (où l'émetteur n'est pas un identifiant), régler une facture, payer avec une carte RFID ou payer en lisant un QR code. L'utilisateur doit s'authentifier avec une double authentification pour effectuer un paiement.
- Pour effectuer un paiement, l'utilisateur doit être connecté. Après être connecté, une double authentification sera lancée avant chaque paiement. L'utilisateur va alors recevoir un email, et une fois que l'utilisateur a cliqué sur le lien contenu dans l'email, il sera connecté.

Système de transaction :

- Toutes les transactions sont visibles sur une table des transactions.
- La table de transactions ne possède que 4 colonnes : émetteur, récepteur, montant et un hash. Le hash est utilisé pour vérifier si la transaction n'a pas été modifiée, puis un nouveau hash est calculé pour chaque transaction en fonction du hash de la transaction précédente et des informations de la transaction. Si le hash ne correspond pas à ces valeurs, cela signifie que la transaction a été modifiée.
- A partir de cette table de transaction, on calcule le solde d'un utilisateur.

Émetteur	Récepteur	Montant	Hash	Date Transaction
Admin	User 1	200	e3b0c	15/12/2025 16h09
Admin	User 2	200	ht84e	15/12/2025 16h10
Admin	User 3	200	l6z5n	15/12/2025 16h11
User 1	User 2	70	s1ny1	16/12/2025 14h35
User 3	User 2	40	q8b2t	17/12/2025 9h05
User 1	User 2	60	v6k5s	17/12/2025 13h45

### Une application mobile :

- L'application mobile permet d'accéder au solde de l'utilisateur.
- L'application permet aussi de payer en scannant son QR code.
- Donne accès à l'historique des transactions récentes.
- Permet de demander le rechargement du solde directement depuis l'application

### Un site web :

- Le site web permet de faire la même chose que l'application.
- Offre la consultation du solde et de l'historique des transactions
- Autorise les paiements via QR code depuis un navigateur web
- Permet une utilisation sur ordinateur pour les utilisateurs ne disposant pas d'un smartphone compatible
- Constitue une alternative pratique et accessible à l'application mobile
- Permet de demander un rechargement du solde en ligne
- Permet aux administrateurs de voir et de gérer la table des transaction et d'effectuer des transactions

### Le raspberry PI :

- Le raspberry PI est la base centrale du système, c'est là où les données sont stockées.
- Héberge les services nécessaires au fonctionnement de la plateforme
- Il fait également office de hotspot afin de faire fonctionner l'application en zone blanche (zone sans réseau).

## Fonctionnalités secondaires

Les fonctionnalités secondaires viennent compléter les fonctionnalités principales afin d'améliorer l'expérience utilisateur, l'ergonomie du système et le suivi des transactions, sans être indispensables au fonctionnement de base de la plateforme.

Pour les utilisateurs :

- Personnalisation de l'interface de l'application (choix de thèmes : clair, sombre, couleurs).
- Consultation détaillée de l'historique des transactions (date, montant, émetteur, récepteur).
- Réception de notifications lors d'un paiement effectué ou reçu.
- Accès simplifié aux dernières transactions depuis la page d'accueil.
- Génération QR codes pour recevoir des paiements.
- Possibilité de supprimer son compte

Pour les commerçants :

- Accès à un tableau de bord récapitulatif.
- Consultation du total des ventes par jour, semaine et mois.
- Visualisation du nombre de transactions effectuées.
- Suivi du nombre de clients uniques.

Pour les admins :

- Tableau de bord global avec graphiques (nombre de transactions, volume total, utilisateurs actifs).
- Accès aux logs système pour le suivi des actions et la détection d'anomalies.

## Fonctionnalités étendues

- Pouvoir payer avec une carte qui est reliée au compte
- Pouvoir exporter les historiques des transactions en pdf
- Pouvoir payer avec le téléphone en NFC
- Centre d'aide intégré (FAQ)
- Pouvoir se connecter avec un compte Google

# Diagrammes

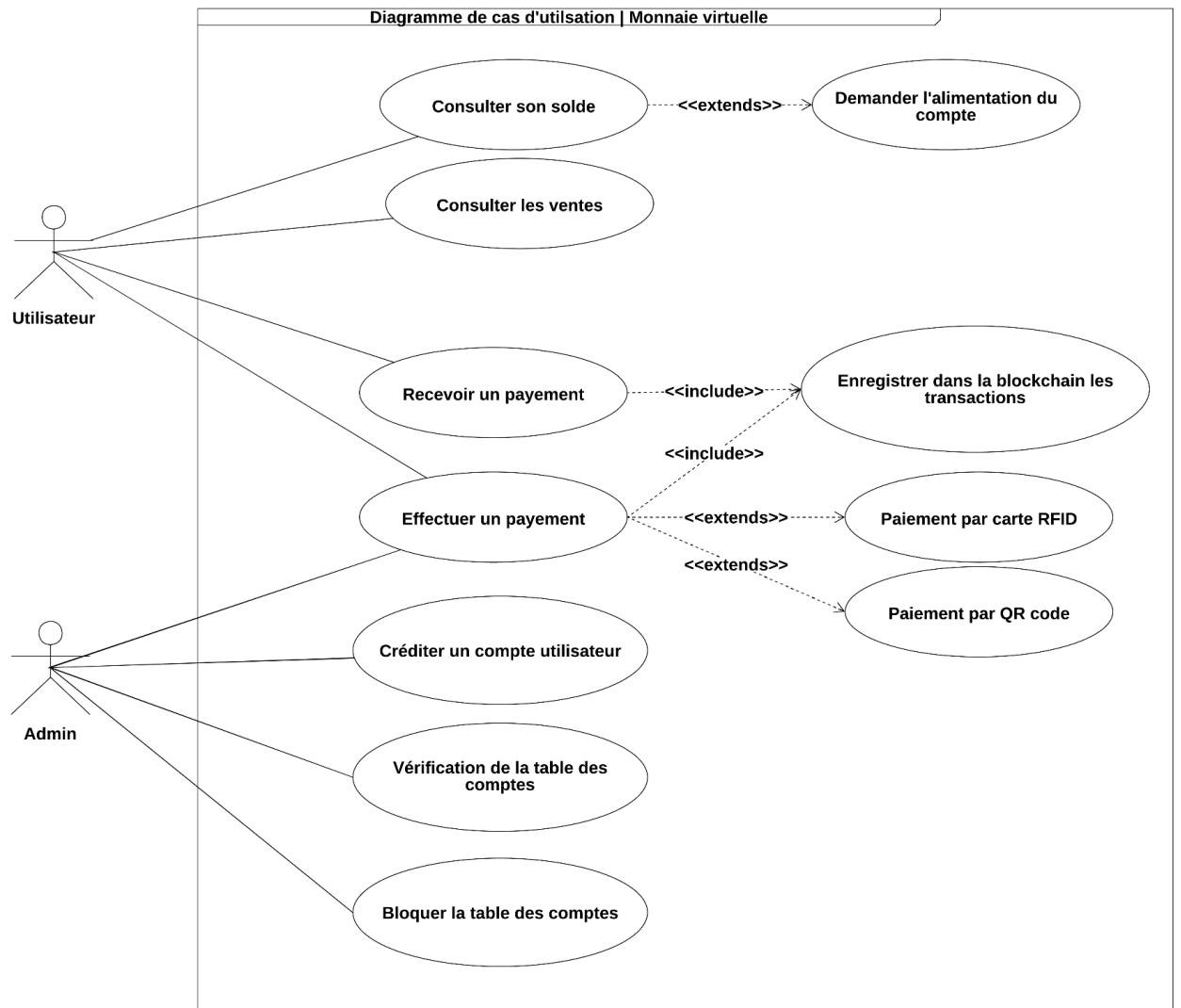


Diagramme d'activité d'une transaction entre 2 utilisateurs

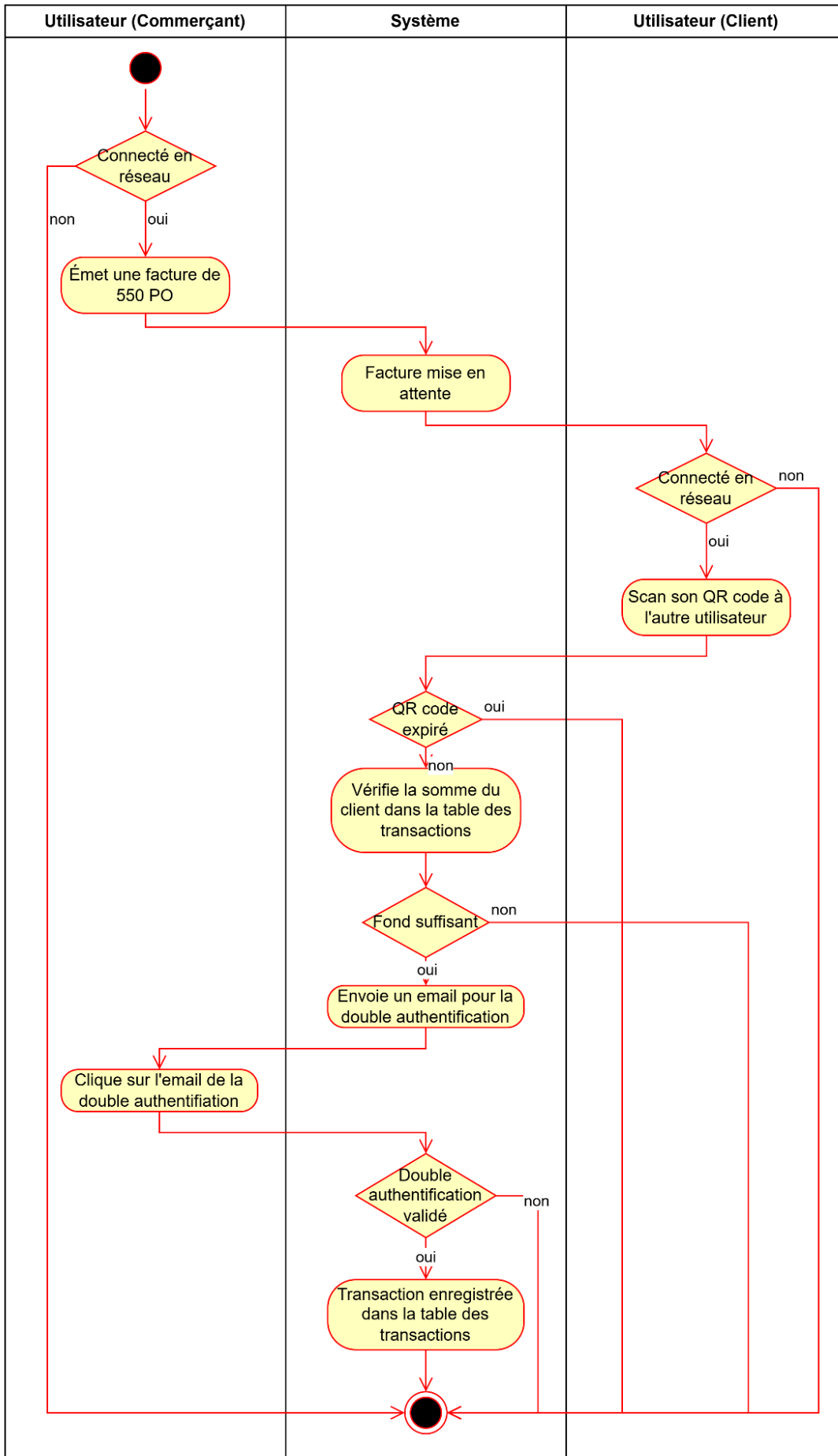


Diagramme d'activité pour l'alimentation d'un compte utilisateur

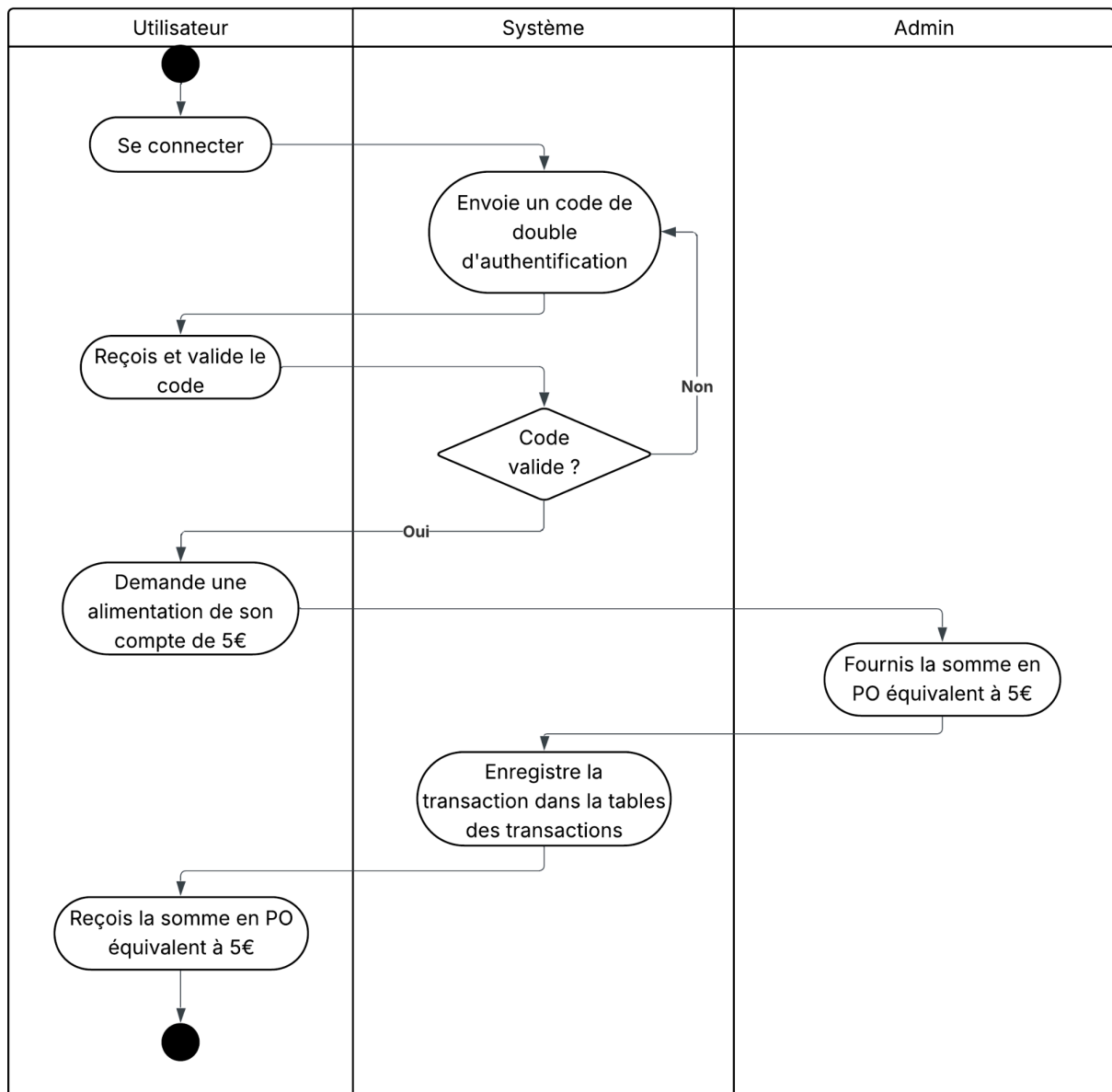
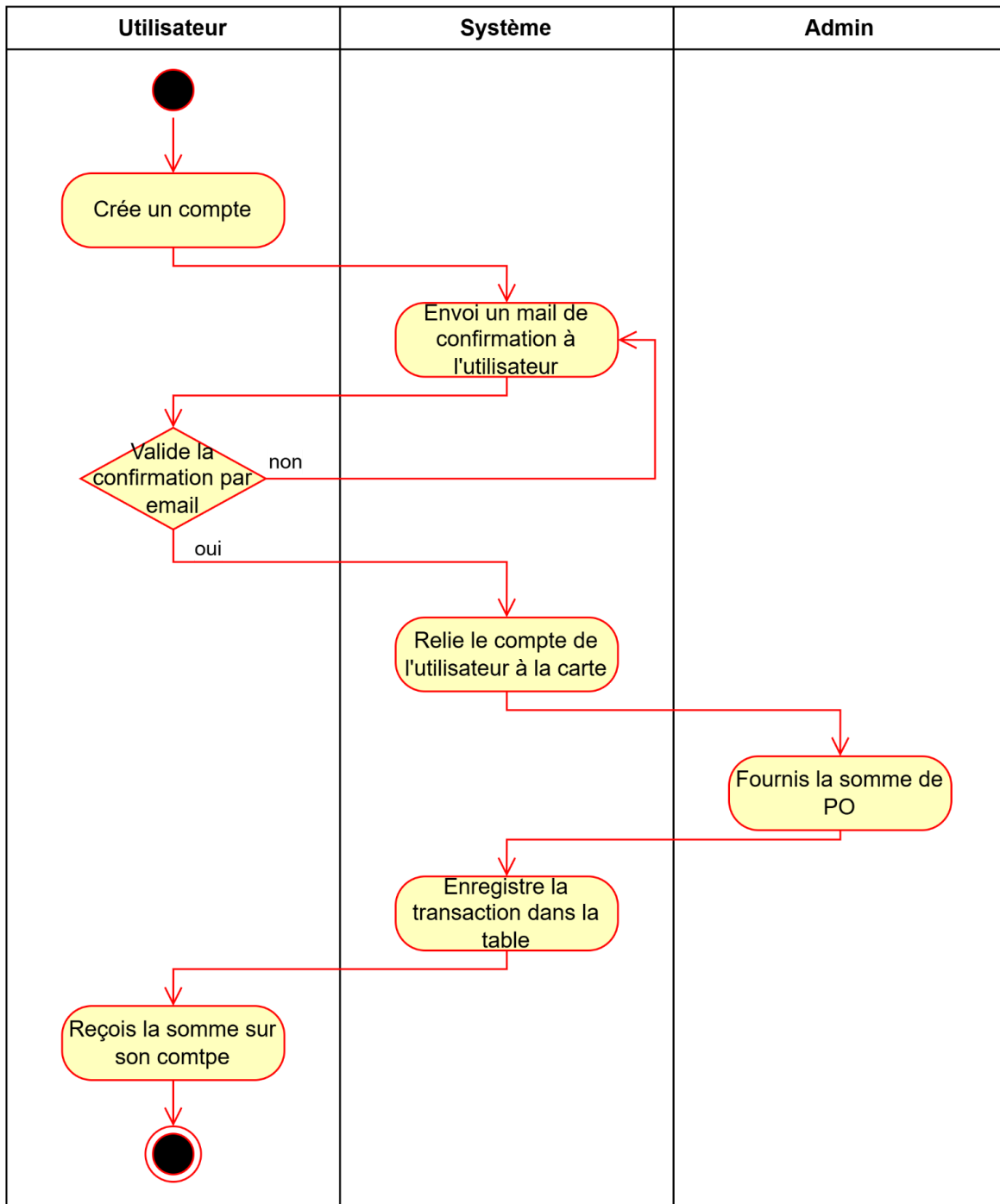


Diagramme d'activité de la création d'un compte utilisateur +  
le chargement d'une carte prépayé en liquide

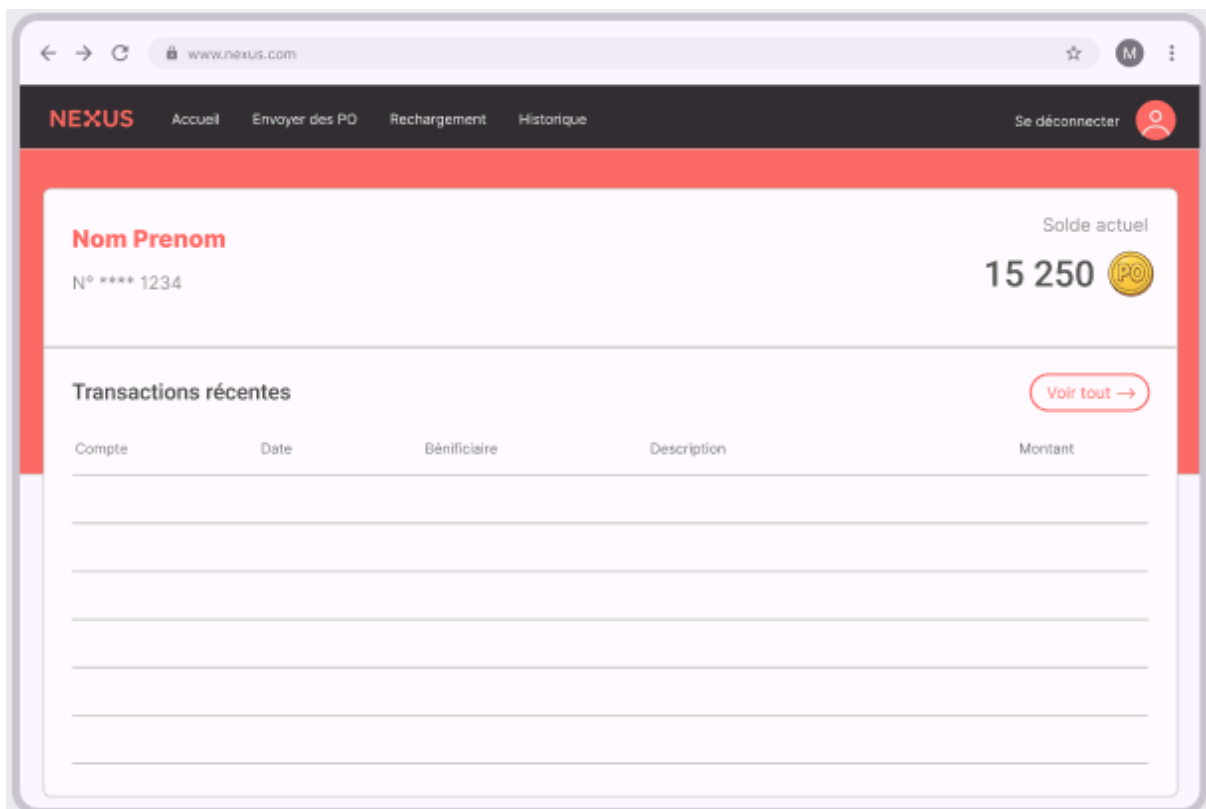
# Maquettes

**Connexion** : Lorsque l'utilisateur lance l'application pour la première fois, il peut créer un compte ou se connecter via divers moyens (identifiant / mot de passe, compte Google...).

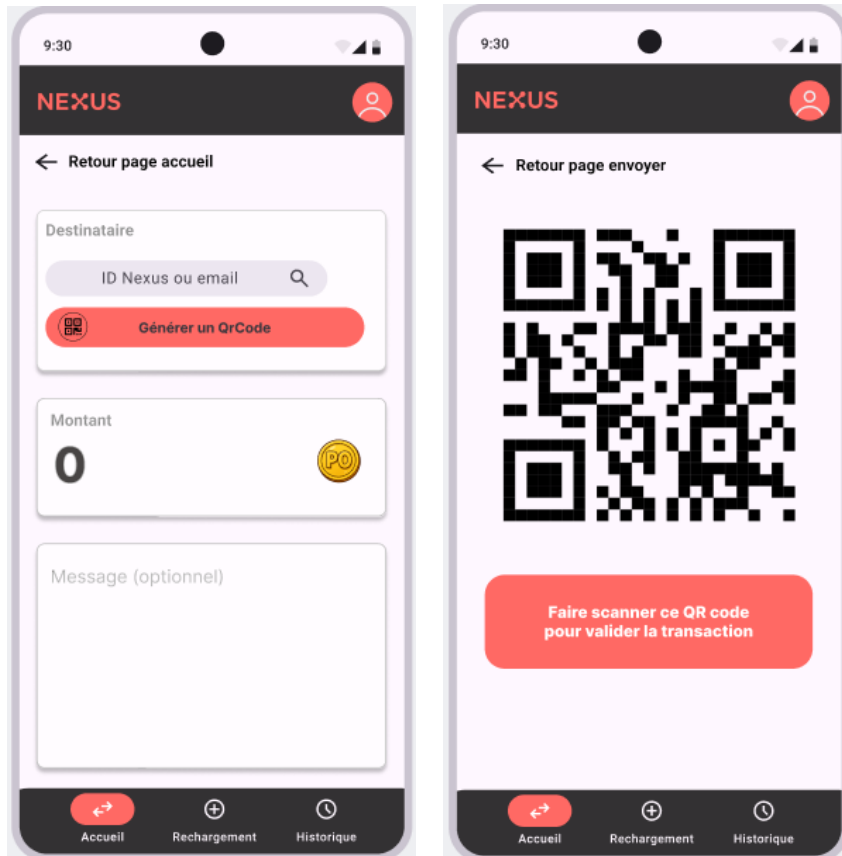




**Page d'accueil :** Cette page sera visible dès la connexion d'un utilisateur. Il peut y retrouver son solde actuel (en PO) et ses transactions récentes. Il a également l'occasion d'envoyer des PO ou de recevoir des PO grâce aux deux boutons disponibles.



**Envoyer des PO** : L'utilisateur doit compléter un formulaire (destinataire, montant, message) afin de fournir les informations utiles à la transaction. Il a également l'opportunité de scanner un QR Code pour gérer ce transfert. Ce QR Code doit être scanné par l'émetteur.



## Base de données

```
CREATE TYPE user_role AS ENUM ('client', 'commercant', 'admin');
```

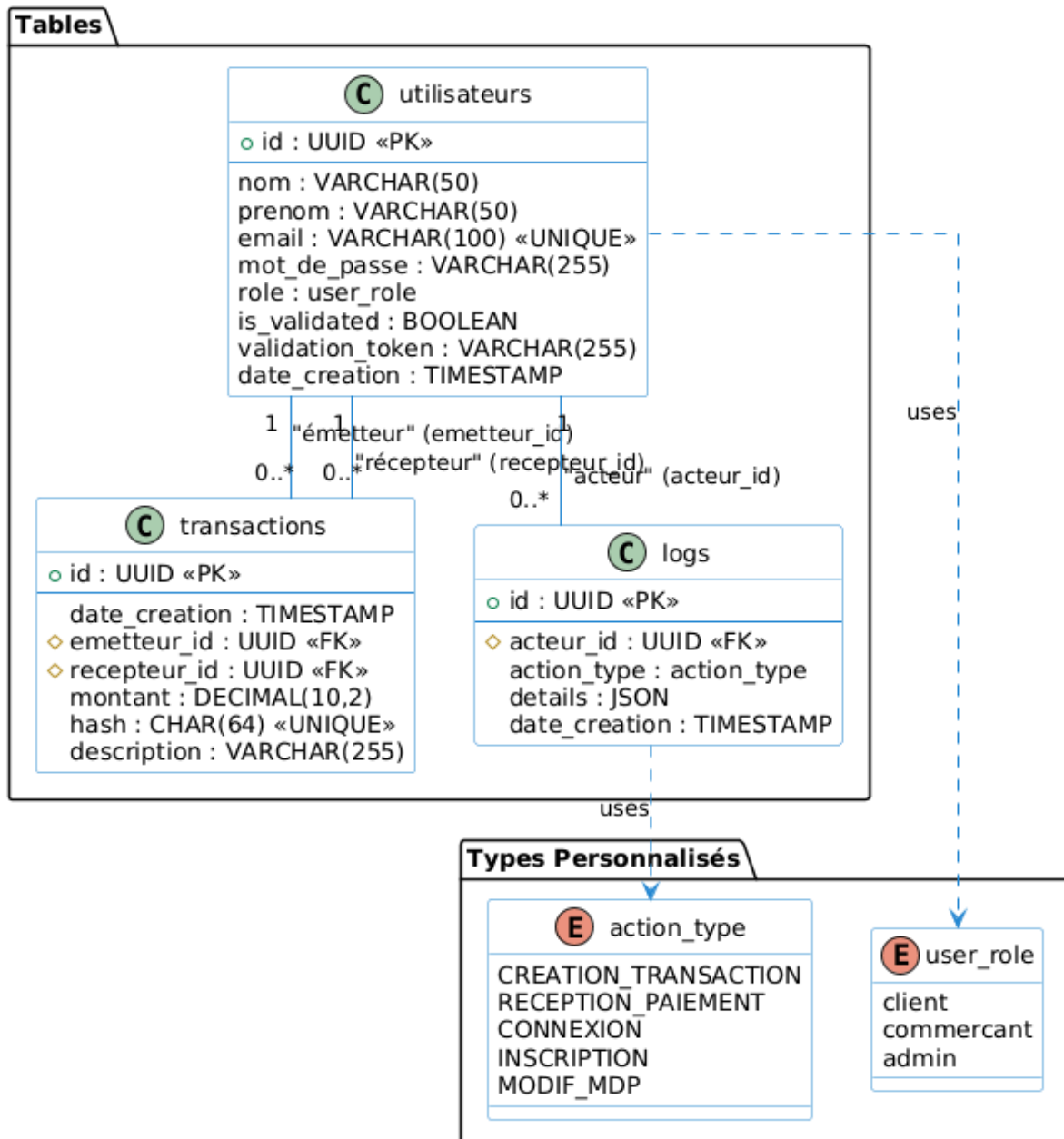
```
CREATE TYPE action_type as ENUM ('CREATION_TRANSACTION',  
'RECEPTION_PAIEMENT', 'CONNEXION', 'INSCRIPTION', 'MODIF_MDP');
```

```
CREATE TABLE utilisateurs (  
    id UUID PRIMARY KEY,  
    nom VARCHAR(50) NOT NULL,  
    prenom VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    mot_de_passe VARCHAR(255) NOT NULL,  
    role user_role DEFAULT 'client',  
    is_validated BOOLEAN DEFAULT false,  
    validation_token VARCHAR(255) DEFAULT NULL,  
    date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE transactions (  
    id UUID PRIMARY KEY,  
    date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    emetteur_id UUID NULL,  
    recepateur_id UUID NULL,  
    montant DECIMAL(10,2) NOT NULL,  
    hash CHAR(64) NOT NULL UNIQUE,  
    description VARCHAR(255) NULL,  
    FOREIGN KEY (emetteur_id) REFERENCES utilisateurs(id) ON  
DELETE SET NULL,  
    FOREIGN KEY (recepateur_id) REFERENCES utilisateurs(id) ON  
DELETE SET NULL  
);
```

```
CREATE TABLE logs (  
    id UUID PRIMARY KEY,  
    acteur_id UUID,  
    action_type action_type,  
    details JSON,  
    date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (acteur_id) REFERENCES utilisateurs(id)  
);
```

```
CREATE INDEX idx_logs_created_at ON logs(date_creation);
```



## Les itérations

- 1<sup>ère</sup> Itération :

La semaine était dédiée à l'analyse des besoins. Une première version d'un compte rendu a été effectuée avec les cas d'utilisation, des diagrammes et des explications sur le fonctionnement du projet. Des débuts de maquette ont été effectués pour le rendu de la version mobile.

- 2<sup>ème</sup> Itération :

Sur la 2<sup>ème</sup> semaine, après une réunion avec notre enseignant sur le projet. On a fait qu'elles que modifications sur le fonctionnement du projet. Les maquettes de la version mobile ont été terminées. On a commencé à réfléchir à la manière dont nous allions mettre en place le projet. Le choix était d'utiliser PHP pour l'api, VueJS pour le front et Flutter pour l'application mobile.

- 3<sup>ème</sup> Itération :

Pour la itération nous avons commencé le développement du projet. On a commencé à faire fonctionner l'api pour avoir quelque route qui marche. Le développement du mobile et du front a aussi commencé en reprenant l'esthétique des maquettes.

- 4<sup>ème</sup> Itération :

Cette itération regroupe plusieurs fonctionnalités effectuées comme la vérification des transactions dans la base de données, suivi d'un blocage de la base de données en cas de changement de données. L'application mobile a aussi été finie, il manquait seulement quelques fonctionnalités. Le front a été presque terminé.

- 5<sup>ème</sup> Itération :

Sur cette dernière semaine,