Application web pour les clients d’un garage

ligne courte

[**1.** Contexte 6](#_Toc100171393)

[1.1. Objectif de la mission 6](#_Toc100171394)

[1.2. Analyse de l’existant 6](#_Toc100171395)

[1.3. Recommandations générales 7](#_Toc100171396)

[1.4. Organisation du document 7](#_Toc100171397)

[**2.** Description générale du futur système 8](#_Toc100171398)

[2.1. Architecture logicielle du système 8](#_Toc100171399)

[2.1.1. Architecture desktop 8](#_Toc100171400)

[2.1.2. Architecture multi-tiers basée sur le web 9](#_Toc100171401)

[2.1.3. Notre recommandation 9](#_Toc100171402)

[2.2. Choix technologiques 10](#_Toc100171403)

[2.2.1. Plateforme de développement 10](#_Toc100171404)

[2.2.2. Système de gestion de base de données (SGBD) 13](#_Toc100171405)

[**1.** **Les SGBD open-source** 13](#_Toc100171406)

[**2.** **Les SGBD payants** 13](#_Toc100171407)

[3. Modules fonctionnels 14](#_Toc100171408)

[**3.1.** Gestion des utilisateurs et des profils 14](#_Toc100171409)

[**3.2.** Gestion des transactions 15](#_Toc100171410)

[**3.3.** Back-office 15](#_Toc100171411)

[**3.4.** Gestion de la collecte 16](#_Toc100171412)

[**4.** Spécifications détaillées des modules 17](#_Toc100171413)

[4.1. Gestion des utilisateurs 17](#_Toc100171414)

[4.1.1 Inscription d’un nouvel utilisateur 17](#_Toc100171415)

[4.1.1.1 Profil concerné 17](#_Toc100171416)

[4.1.1.2 Description fonctionnelle 17](#_Toc100171417)

[4.1.2 Connexion d’un utilisateur 18](#_Toc100171418)

[4.1.2.1 Profil concerné 18](#_Toc100171419)

[4.1.2.2 Description fonctionnelle 18](#_Toc100171420)

[4.1.3 Consultation du profil 18](#_Toc100171421)

[4.1.3.1 Profil concerné 18](#_Toc100171422)

[4.1.3.2 Description fonctionnelle 19](#_Toc100171423)

[4.1.4 Modification des informations d’un utilisateur 19](#_Toc100171424)

[4.1.5 Extraits copies d’écran 19](#_Toc100171425)

[22](#_Toc100171426)

[4.2. Gestion des transactions 22](#_Toc100171427)

[4.2.1 Dépôt 22](#_Toc100171428)

[4.2.1.1 Profil concerné 23](#_Toc100171429)

[4.2.1.2 Description fonctionnelle 23](#_Toc100171430)

[4.2.2 Retrait 23](#_Toc100171431)

[4.2.2.1 Profil concerné 23](#_Toc100171432)

[4.2.2.2 Description fonctionnelle 23](#_Toc100171433)

[4.2.3 Consultation du solde 24](#_Toc100171434)

[4.2.3.1 Profil concerné 24](#_Toc100171435)

[4.2.3.2 Description fonctionnelle 24](#_Toc100171436)

[4.2.4 Consultation de l’historique des transactions 24](#_Toc100171437)

[4.2.4.1 Profil concerné 25](#_Toc100171438)

[4.2.4.2 Description fonctionnelle 25](#_Toc100171439)

[4.2.5 Transfert 26](#_Toc100171440)

[4.2.5.1 Profil concerné 26](#_Toc100171441)

[4.2.5.2 Description fonctionnelle 26](#_Toc100171442)

[4.2.5.2.1 Formulaire de transfert 26](#_Toc100171443)

[4.2.5.2.2 Validation par code secret 27](#_Toc100171444)

[4.2.6 Extraits de copies d’écran 27](#_Toc100171445)

[4.3 Back-office 32](#_Toc100171446)

[4.3.1. Collection d’argent liquide par kiosque 32](#_Toc100171447)

[4.3.1.1. Profil concerné 32](#_Toc100171448)

[4.3.1.2. Description fonctionnelle 32](#_Toc100171449)

[4.3.2. Suivi des kiosques 32](#_Toc100171450)

[4.3.2.1 Suivi des transactions 32](#_Toc100171451)

[4.3.2.1.1. Profil concerné 32](#_Toc100171452)

[4.3.2.1.2. Description fonctionnelle 32](#_Toc100171453)

[4.3.2.2 Suivi de l’argent liquide par kiosque 32](#_Toc100171454)

[4.3.2.2.1 Profil concerné 32](#_Toc100171455)

[4.3.2.2.2. Description fonctionnelle 32](#_Toc100171456)

[4.3.2.3 Limite de collection par kiosque 32](#_Toc100171457)

[4.3.2.3.1. Profil concerné 32](#_Toc100171458)

[4.3.2.3.2. Description fonctionnelle 32](#_Toc100171459)

[4.3.3. Extrait de copies d’écran 32](#_Toc100171460)

[4.4 Gestion de la collecte 33](#_Toc100171461)

[4.4.1. Assignation d’un kiosque à un collecteur 33](#_Toc100171462)

[4.4.1.1. Profil concerné 33](#_Toc100171463)

[4.4.1.2. Description fonctionnelle 33](#_Toc100171464)

[4.4.2. L’état d’une collecte 33](#_Toc100171465)

[4.4.2.1. Profil concerné 33](#_Toc100171466)

[4.4.2.2. Description fonctionnelle 33](#_Toc100171467)

[4.4.3. Notification de collection effectuée 33](#_Toc100171468)

[4.4.3.1. Profil concerné 33](#_Toc100171469)

[4.4.3.2. Description fonctionnelle 33](#_Toc100171470)

[4.4.4. Le dépôt de la collecte à la plateforme 33](#_Toc100171471)

[4.4.4.1. Profil concerné 33](#_Toc100171472)

[4.4.4.2. Description fonctionnelle 33](#_Toc100171473)

[4.4.5. Accusé de réception 33](#_Toc100171474)

[4.4.5.1. Profil concerné 33](#_Toc100171475)

[4.4.5.2. Description fonctionnelle 33](#_Toc100171476)

[4.4.6. Extraits de copies d’écran 33](#_Toc100171477)

[**5.** Infrastructure et sécurité 33](#_Toc100171478)

[**6.** Conclusion 33](#_Toc100171479)

# Contexte

## Objectif de la mission

Soucieux de ne pas rater le train de la digitalisation à Madagascar, nous avons lancé la mission du MGA Pay, dans le but de faciliter la transaction monétaire à Madagascar et ainsi, diminuer l’impression de billets au niveau de la Banque Centrale.

Ainsi, les objectifs principaux de la mission étaient de :

* Effectuer une analyse de l’existant,
* Proposer des recommandations sur :
* Les applicatifs informatiques à mettre en place
* Les procédures à adopter

Ce document concerne les spécifications fonctionnelles du futur système ainsi que les recommandations sur l’infrastructure et les réseaux.

## Analyse de l’existant

Aujourd’hui, les frais du Mobile Banking sont très coûteux, à Madagascar. Les transactions monétaires sont encore difficiles.

Il en résulte qu’il est très difficile, surtout au niveau des utilisateurs des services de Mobile Banking, actuellement opérationnels, de manipuler avec autonomie, à faible coût et en toute sécurité, les transactions monétaires. Aussi, plus de 3 millions d’internautes malgaches utilisent Facebook quotidiennement. Cela constitue un atout majeur dans la mise en œuvre du projet MGA Pay.

## Recommandations générales

Eu égard à :

* L’insécurité des données monétaires actuellement,
* Aux coûts élevés des frais du Mobile Banking
* La diminution de la valeur de l’Ariary

Nous recommandons la mise en place d’un nouveau mode de paiement sécurisé, à moindre coût, grâce à une plateforme facile à manipuler, accessible, intégrée et développée autour de la technologie web.

L’accès à cette application sera protégé par un login et mot de passe. Chaque utilisateur aura accès aux fonctionnalités accordées par rapport à son profil.

## Organisation du document

Ce document concerne les spécifications fonctionnelles du futur système ainsi que les recommandations sur l’infrastructure et réseaux.

La partie qui suit, contiendrait une description succincte de l’ensemble de nos recommandations.

Ensuite, chaque module du futur système fera l’objet d’une section dédiée et y seront présentées les différentes fonctionnalités à mettre en place.

La partie à propos de nos recommandations sur l'infrastructure et les réseaux précède notre conclusion pour clôturer ce document.

Toutefois, des éléments pouvant apporter plus d'éclaircissement seront présentés en annexe (exemple de wireframe, scan des divers documents, …).

# Description générale du futur système

## Architecture logicielle du système

De nos jours, 2 options d'architecture sont mises à notre disposition. Nous avons d’une part celle dite desktop et d’autre part celle multi-tiers basée sur le Web.

### Architecture desktop

Une application construite autour de ce modèle n’est destinée qu’à une plateforme unique bien définie (OS X, Linux, Windows, etc.). Cette façon de procéder implique donc nécessairement une installation de l’application pour chaque ordinateur client. Ces types d’applications sont développées avec la technologie comme Visual Basic, Windev, etc.

a) **Avantages**

* Généralement, cette architecture offre une plus forte interactivité au niveau de l’interface graphique.
* Du fait que l’application a été conçue pour une plateforme spécifique, elle a donc accès aux fonctionnalités natives de cette dernière.

b) **Inconvénients**

* La maintenance peut occasionner des soucis d’incompatibilités lors :
  + d'une mise à jour de l’OS en question (ex : passer de Windows 10 à Windows 11)
  + d’un changement de plateforme (ex : passer de OS X à Linux).

Il faudra alors revoir en intégralité l’application avant de pouvoir l’utiliser de nouveau.

* Il faudra également développer plusieurs versions de la même application pour la rendre accessible sur plusieurs plateformes (ex : une version pour mobiles (Android, iOS), une autre pour Windows, etc.).

### Architecture multi-tiers basée sur le web

La principale différence ergonomique avec l’architecture présentée plus haut réside dans le fait qu’il n’y a aucun logiciel à installer pour les clients. Ces derniers n’auront besoin que d’un navigateur Web, normalement fourni avec le plateforme, qui serait alors la porte d’entrée de l’application. Ainsi, elle deviendrait multiplateforme.

En optant pour cette architecture, le développement de l’application se fera sur plusieurs couches, notamment :

* Une couche d’accès aux données, responsables des opérations vers le système de gestion de la base des données,
* Une couche métier, qui sera en charge de l’implémentation des logiques métiers et les règles de gestion du système,
* Et une couche de présentation.

a) **Avantages**

* L’application est accessible n’importe où à tout moment tant que le client est connecté aux réseaux locaux de l’entreprise ou à l’Internet (sous réserve de la mise en place des dispositifs de sécurité adéquate),
* La maintenance de l’application est plus aisée.

b) **Inconvénient**

* Son développement est plus chronophage étant donné qu’il sera forcément sur plusieurs couches.

### Notre recommandation

Malgré l’interactivité appréciable proposée par l’architecture desktop, nous recommandons d’opter pour la mise en place d’une application multi-tiers orienté web.

En effet, nous jugeons qu’il serait préférable que les informations soient plus accessibles et également que l’application soit plus indépendante du plateforme où elle est utilisée.

La future application sera disponible sur une interface web par l’intermédiaire d’un onglet du menu de Facebook.

## Choix technologiques

Les choix des technologies nécessaires à la réalisation de l’application s’avèrent très importantes dans la réussite d’une telle mission.

### Plateforme de développement

Ces plateformes sont principalement constituées de frameworks avec le ou les langage(s) de programmation pour leur exploitation. Nous distinguons parmi les plus en vogues dans le monde professionnel :

1. **Fullstack Javascript**

De nos jours, Javascript est devenu un langage de programmation réputée notamment grâce à l’émergence de différents frameworks, tels que Node.JS pour la partie back-end et Angular et React assurant le coté front-end pour ne citer qu’eux, qui forment ensemble l’univers Javascript.

1. **Avantages**

* La rapidité est remarquable.
* L’unicité du langage de programmation, à savoir Javascript, est aussi un atout considérable pour les développeurs de l’application.

1. **Inconvénient**

* Il s’agit d’une technologie jeune ce qui signifie qu’elle n'a pas encore fait ses preuves dans des applications d’entreprise. Par exemple, elle ne prend pas en charge la création d’applications transactionnelles.

1. **PHP et ses frameworks**

Le PHP est un des langages de développement d’applications les plus connus dans le monde du Web dynamique.

* 1. **Avantages**
* Cette plateforme a été conçue principalement pour les sites d’informations et les sites vitrines ce qui lui a valu sa réputation.
  1. **Inconvénient**
* ​​​​Elle est inadaptée pour les applications de grande envergure telles que celles d’entreprises malgré l’apparition de frameworks plus matures comme Symfony ou Laravel.

1. **.NET**

Quant à .NET, il a été développé par Microsoft Corporation dans le but de répondre aux exigences des grandes entreprises en termes de développement d’applications.

1. **Avantages**

* C’est une plateforme mature, rapide, performante, maintenable et stable, tout ce dont la réalisation d’une application destinée aux grandes firmes requiert.

1. **Inconvénient**

* Il y a d’abord son coût financier vu qu’il faut payer une licence logicielle pour pouvoir en faire usage.
* Du côté serveur, nous sommes dans l’obligation de nous servir de Windows Server ; c’est-à-dire qu’il est impossible de l’utiliser sans s’appuyer sur les technologies de Microsoft Corporation.

1. **Java et ses frameworks**

Les technologies Java ne sont plus à présenter tellement elles sont omniprésentes. En plus d’être un langage de programmation très connu avec une grande communauté de développeurs, Java c’est également un ensemble de solutions proposées pour l’élaboration d’applications à but professionnel.

1. **Avantages**

* Il n’a rien à envier à .NET en termes de fiabilité en plus d’être open-source contrairement à son homologue de chez Microsoft Corporation.
* Puisque cette technologie est implantée dans le monde du Web depuis ses débuts, les ressources la concernant abondent. Cela est également dû aux nombreux experts de cette dernière l’utilisant au quotidien. Elle évolue donc sans cesse.
* Poursuivons notre comparaison avec .NET, Java dispose de ce que nous appelons une « machine virtuelle » ce qui lui procure la capacité d’être multiplateforme.

1. **Inconvénient**

* Comme nous l’avons vu, Java est très complet certes cependant son point faible demeure dans sa complexité. Il est très difficile à appréhender pour un non-initié.

Notre recommandation

Pour le bien du projet, il serait plus judicieux d’opter pour la plateforme Java qui cumule le plus de points forts tout en ayant des points faibles facilement remédiables.

### Système de gestion de base de données (SGBD)

Il est à préciser qu’il s’agit également d’un élément important pour réussir à concevoir une application vu qu’il persiste toutes les informations et se charge d’assurer l’intégrité des données. C’est pour cela que la performance de l’application globale se base surtout sur sa gestion des données, en d’autres termes, de son SGBD. Nous pouvons en considérer 2 grands types :

### **Les SGBD open-source**

Parmi ces derniers, les plus référencés et appréciés sont MySQL et PostgreSQL qui sont des SGBD dits « relationnels ».

1. **Avantages**

* Ils suffisent largement pour des applications assez sophistiquées étant donné que ces SGBD sont complets, du moins pour leur prix.

1. **Inconvénients**

* Nous nous rendons vite compte de leurs limites lors de la conception d'applications plus complexes et avancées. Effectivement, certaines fonctionnalités sont soit complexes à mettre en œuvre soit inexistantes.

### **Les SGBD payants**

Dans le monde industriel, les plus célèbres pour leur robustesse et leur performance sont Oracle Database et Microsoft SQL Server. Ils sont également des SGBD « relationnels ».

1. **Avantages**

* Ils sont à la fois stables et très performants.

1. **Inconvénients**

* Aujourd’hui, il n’y a pas grand-chose à leur reprocher à part leur coût d’exploitation.

Notre recommandation

Back-end : Java Spring Boot

Front-end: Angular

Bdd: Postgres

## Modules fonctionnels

L’application web recommandée aura les caractéristiques ci-après :

* L’accès sera protégé par un nom d'utilisateur (login) et un mot de passe,
* Elle sera composée de plusieurs modules qui comportera plusieurs fonctionnalités (les détails seront dans la partie qui suit),
* L’application devra être accessible à partir du menu de l’application Facebook et adoptera la technique de “responsive design” garantissant l’affichage sur tous les écrans mobile ou desktop.

### Gestion des utilisateurs et des profils

Ce module est dédié à la gestion des utilisateurs de la plateforme, ainsi que des profils qui peuvent accéder aux différentes fonctionnalités. Il est à noter qu’un accès sécurisé sera créé afin de pouvoir accéder aux différentes fonctionnalités sur la plateforme.

Il y aura 4 types de profils :

* Client
* Gestionnaire de kiosque
* Collecteur
* Administrateur

Par conséquent, chaque utilisateur aura sa propre authentification (login / mot de passe) pour accéder à son compte, et chaque profil a des fonctionnalités attribuées.

### Gestion des transactions

Ce module prend en charge les mouvements du compte bancaire de l’utilisateur, à savoir :

* Le retrait
* Le dépôt
* Le transfert

Entre autres, on peut aussi consulter :

* Le solde actuel du compte
* L’historique des transactions antérieures (retrait, dépôt et transfert compris)

Ce module permettra aux utilisateurs de suivre et vérifier en temps réel les mouvements de son compte bancaire effectué sur le plateforme MGA Pay, mais aussi, le solde en cours dans son compte physique, à la banque.

### Back-office

En utilisant le solde des dépôts, retraits et transfert effectués par tous les utilisateurs, on peut établir une analyse des chiffres sous-forme de graphiques et/ou de tableau.

On peut y retrouver :

* Les statistiques de nombres d’utilisateurs par rapport au temps
* La liste des utilisateurs inscrits à la plateforme et leurs informations importantes
* Le total des dépôts des utilisateurs par rapport au temps (ou entre 2 dates)
* Le total des retraits des utilisateurs par rapport au temps (ou entre 2 dates)
* Le total des transferts des utilisateurs par rapport au temps (ou entre 2 dates)

On peut aussi y trouver plusieurs suivis, à savoir :

* Suivi de la collection d’argent liquide, pour chaque kiosque
* Suivi de chaque kiosque par rapport à la collection, au solde et à l’argent liquide supposément à sa disposition

### Gestion de la collecte

Ce module permettra de suivre et vérifier en temps réel la collecte d’argent liquide auprès des kiosques.

On peut y trouver :

* L’assignation d’un kiosque à un collecteur
* L’état d’une collecte
* Le dépôt de la collecte à la plateforme

# Spécifications détaillées des modules

## 4.1. Gestion des utilisateurs

### 4.1.1 Inscription d’un nouvel utilisateur

Cette fonctionnalité permet d’ajouter un nouvel utilisateur au plateforme. Ce nouvel utilisateur peut avoir à son nom, un compte à la banque ou non.

#### 4.1.1.1 Profil concerné

* Client

#### 4.1.1.2 Description fonctionnelle

Si l’utilisateur a déjà un compte bancaire, il s’inscrit simplement à la plateforme en renseignant les champs suivants :

* Nom
* Prénom
* Nom d’utilisateur
* Numéro de compte
* Numéro de téléphone/ email
* Mot de passe
* Confirmation du mot de passe

Sinon, s’ajoute à ceux-là, la liste des pièces à fournir suivante, en pièces jointes (scan, format pdf):

* Photocopie de CIN légalisée
* Certificat de résidence

### 4.1.2 Connexion d’un utilisateur

Cette fonctionnalité permet à l’utilisateur, déjà inscris à la plateforme, d’accéder à son compte et aux différentes actions possibles à faire.

#### 4.1.2.1 Profil concerné

* Client
* Gestionnaire de kiosque
* Administrateur
* Collecteur

#### 4.1.2.2 Description fonctionnelle

Les champs à renseigner sont :

* Nom d’utilisateur
* Mot de passe

Si, après vérifications, les informations entrées correspondent à un compte, l’utilisateur peut y accéder.

### Consultation du profil

Cette interface regroupe toutes les informations qui concernent l’utilisateur connecté, ainsi qu’une modification au niveau de sécurité de son compte.

#### Profil concerné

* Client
* Kiosque
* Collecteur

#### Description fonctionnelle

Elle réunit les renseignements suivantes :

* Le nom de l’utilisateur
* Les prénoms de l’utilisateur
* Sa date de naissance
* Son adresse
* Son numéro de téléphone
* Son nom sur Facebook
* Son numéro de carte d’identité

Mais elle permet aussi à l’utilisateur d’effectuer des changements au niveau de son mot de passe au cas où son compte passe entre de mauvaises mains par exemple.

### Modification des informations d’un utilisateur

Cette fonctionnalité permet à l’utilisateur d’effectuer des changements au niveau de son mot de passe au cas où son compte passe entre de mauvaises mains par exemple.

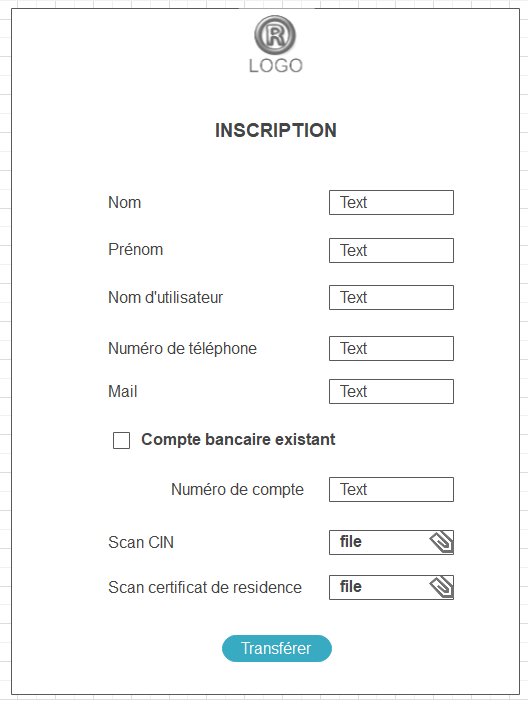
### Extraits copies d’écran

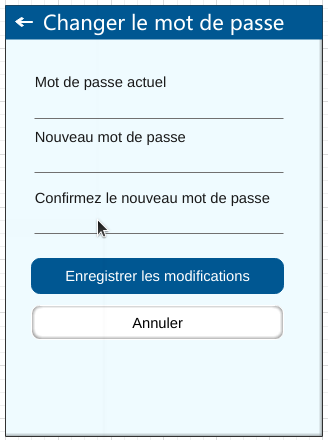
* Connexion d’utilisateur

Une image contenant texte

Description générée automatiquement

* Inscription



* Changement de mot de passe
* Consultation du profil

## 

## 4.2. Gestion des transactions

### 4.2.1 Dépôt

Ce module est conçu pour les kiosques. Puisque contrairement à l’utilisateur normal, les kiosques peuvent effectuer les dépôts sur le compte des autres utilisateurs.

Ainsi lorsque le kiosque valide le dépôt, un message de confirmation lui sera adressé si oui ou non, il veut vraiment exécuter la demande pour le montant et au destinataire saisi ou s’il y a eu une erreur de frappe ou d’inattention.

Cette confirmation se fera par la saisie de son mot de passe puis en appuyant sur le bouton « Valider ». En cas de refus, l’utilisateur appuiera sur le bouton « Annuler »

#### 4.2.1.1 Profil concerné

* Gestionnaire de kiosque
* Client

#### 4.2.1.2 Description fonctionnelle

Pour ce faire, le kiosque devra renseigner les champs suivants :

* Le destinataire : le compte de l’utilisateur auquel nous voulons effectuer le dépôt. Ce champ sera rempli soit par son numéro de téléphone soit par son adresse électronique
* Le montant : le montant à déposer sur le compte du destinataire en Ariary

### 4.2.2 Retrait

Ce module prend en charge le retrait bancaire. Il permettra aux utilisateurs de retirer des espèces de son compte bancaire au moyen de son profil inscrit dans le plateforme qui est relié à son compte bancaire. Le frais de retrait est gratuit jusqu’à 10 000 Ar et l’utilisateur peut faire autant de retrait dans une journée à partir d’un montant de 2000 Ar.

#### 4.2.2.1 Profil concerné

* Client
* Gestionnaire de kiosque

#### 4.2.2.2 Description fonctionnelle

Dans ce module, les informations suivantes seront à enregistrer :

* Montant : ce champ sert à enregistrer le montant à retirer dans le compte de l’utilisateur
* Kiosque : ce champ indique la référence du kiosque où l’utilisateur a fait son retrait
* Mot de passe : ce champ enregistre le mot de passe de l’utilisateur pour confirmation de l’effective du retrait

### 4.2.3 Consultation du solde

Cette fonctionnalité permet à l’utilisateur de vérifier et consulter le montant de son solde bancaire, en temps réel, par rapport aux transactions qu’il a déjà effectué.

#### 4.2.3.1 Profil concerné

* Client
* Gestionnaire de kiosque

#### 4.2.3.2 Description fonctionnelle

La consultation du solde se fera après une confirmation, par le biais, de la saisie du mot de passe :

* Le montant du solde actuel
* La référence de la demande
* La consigne pour recevoir le solde par sms ou email

### Consultation de l’historique des transactions

Ce module concerne l’historique des transactions de l’utilisateur. Il permet de connaître et de retracer toutes les opérations enregistrées sur son compte, au crédit et au débit et aussi permet également un suivi de sa situation budgétaire et pouvoir réagir rapidement en cas d’erreur ou d’opérations frauduleuses. Les transactions enregistrées au cours des 3 derniers mois sont affichés par défaut mais peuvent être filtrées.

#### Profil concerné

* Client
* Gestionnaire de kiosque

#### Description fonctionnelle

Ce module est divisé en plusieurs affichages :

* Affichage des retraits
* Affichage des dépôts
* Affichage des transferts

Un champ de filtre sera présent dans le module pour que l’utilisateur puisse voir les transactions de son choix à partir du mois de son choix.

* Affichage des retraits

Les informations suivantes seront affichées :

* + Référence de la transaction
  + Date et heure de la transaction
  + Le montant du retrait
  + Référence du kiosque et numéro de téléphone du kiosque
  + Frais de la transaction
  + Le solde final après le retrait
* Affichage des dépôts
  + Référence de la transaction
  + Date et heure de la transaction
  + Le montant du dépôt
  + Référence et numéro de téléphone du kiosque
  + Le solde final après le dépôt
* Affichage des transferts
  + Référence de la transaction
  + Date et heure de la transaction
  + Le montant du transfert
  + Nom compte et numéro de téléphone du destinataire
  + Frais de la transaction
  + Raison du transfert
  + Solde final après le transfert

### Transfert

Le transfert d’argent est l’un des modules clés de l’application. Il permet à un client de faire une transaction inter-utilisateur. Plus précisément, un transfert d’argent d’un compte à un autre. Cela tout en suivant les règles de transfert dictées par la banque.

#### Profil concerné

* Client
* Gestionnaire de kiosque

#### Description fonctionnelle

Ce module comprend essentiellement les fonctionnalités suivantes :

* Formulaire de transfert
* Validation par code secret

Le transfert d’argent doit être validé par le mot de passe du client. Après cela, il recevra une notification par sms et par mail.

##### Formulaire de transfert

Ce formulaire comporte les champs suivants :

* Nom d’utilisateur du destinataire
* Description du transfert
* Montant à transférer

##### Validation par code secret

Afin de confirmer le transfert d’argent vers l’autre client, la saisie du mot de passe est requise. Ainsi, on verra sur cet affichage :

* Un récapitulatif du transfert
* Un champ pour le la saisie du mot de passe

### 4.2.6 Extraits de copies d’écran

* Consultation de solde

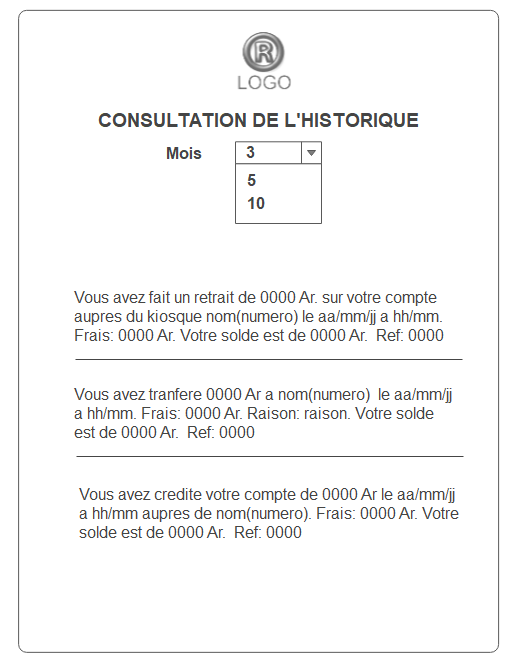
Une image contenant texte

Description générée automatiquement

* Confirmation par mot de passe



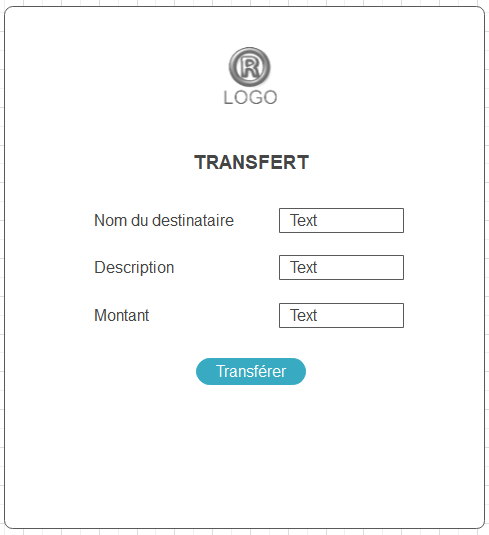
* Consultation de l’historique de transaction



* Retrait



* Transfert



* Dépôt



## Back-office

### 4.3.1. Collection d’argent liquide par kiosque

#### 4.3.1.1. Profil concerné

#### 4.3.1.2. Description fonctionnelle

### 4.3.2. Suivi des kiosques

#### 4.3.2.1 Suivi des transactions

##### 4.3.2.1.1. Profil concerné

##### 4.3.2.1.2. Description fonctionnelle

#### 4.3.2.2 Suivi de l’argent liquide par kiosque

##### 4.3.2.2.1 Profil concerné

##### 4.3.2.2.2. Description fonctionnelle

#### 4.3.2.3 Limite de collection par kiosque

##### 4.3.2.3.1. Profil concerné

##### 4.3.2.3.2. Description fonctionnelle

### 4.3.3. Extrait de copies d’écran

## Gestion de la collecte

### 4.4.1. Assignation d’un kiosque à un collecteur

#### 4.4.1.1. Profil concerné

* Administrateur

#### 4.4.1.2. Description fonctionnelle

Ce module permet à l’administrateur d’assigner un collecteur à un kiosque. Seul le collecteur attribué pourra effectuer le retrait de l’argent liquide au niveau du kiosque. L’assignation ne sera que provisoire, c’est-à-dire, aucune attribution ne sera fixe, pour une question de sécurité et d’intégrité.

L’assignation de kiosque aux collecteurs se fait aussi de manière aléatoire. Ce qui veut dire que l’administrateur ne choisit pas quel collecteur envoyer vers un kiosque particulier. Le système choisira le kiosque et le collecteur au hasard et les associera, de manière aléatoire.

Après avoir générer l’assignation par le biais d’un bouton, on verra s’afficher, en dessous, les informations suivantes :

* La localisation du kiosque
* L’identifiant du gestionnaire de kiosque
* Le montant à collecter
* L’identifiant du collecteur attribué

Le ou les administrateurs de la plateforme ne pourrons avoir accès au profil du collecteur que si la date de collection dépasse un certain seuille (une semaine, un mois ou autre).

Comme chaque module, toutes les actions à risques sont protégées par une confirmation par mot de passe.

### 4.4.2. L’état d’une collecte

#### 4.4.2.1. Profil concerné

* Administrateur

#### 4.4.2.2. Description fonctionnelle

Cette fonctionnalité permet à l’administrateur de suivre, en temps réel, l’état de la collecte au niveau de chacun des kiosques.

On peut donc y trouver :

* Le centre du kiosque (localisation)
* L’état de la collecte (collectée ou non)
* L’identifiant du collecteur attribué
* Le montant à collecter

La suivi de l’état de la collecte se fait, soit, hebdomadairement, mensuellement, ou selon la convenance.

### 4.4.3. Notification de collection effectuée

#### 4.4.3.1. Profil concerné

#### 4.4.3.2. Description fonctionnelle

### 4.4.4. Le dépôt de la collecte à la plateforme

#### 4.4.4.1. Profil concerné

#### 4.4.4.2. Description fonctionnelle

### 4.4.5. Accusé de réception

#### 4.4.5.1. Profil concerné

#### 4.4.5.2. Description fonctionnelle

### 4.4.6. Extraits de copies d’écran

* Suivi de l’état de la collecte
* Assignation d’un kiosque à un collecteur

Une image contenant table

Description générée automatiquement

# Infrastructure et sécurité

# Conclusion