

**REPUBLIQUE DU CAMEROUN Paix-Travail-Patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT AFRICAIN D’INFORMATIQUE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-Travail-Patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT AFRICAIN D’INFORMATIQUE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

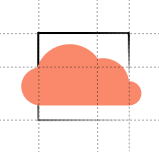
\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*

**REPUBLIQUE DU CAMEROUN Paix-travail-patrie** \*\*\*\*\*\*\*\* **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

\*\*\*\*\*\*\*\*



RAPPORT DE STAGE ACADEMIQUE

THEME : Mise en place d’une plateforme web et mobile de réservation de billets de transport urbain

**Académique de :**

Fenkam Jules Blaise

(Administrateur et intégrateur des SI)

Année Académique : 2022-2023

**Professionnel de :**

Nkeng Kenneth Johan

(PhD in MECHATRONICS)

Rédigé et soutenus par :

Fogue Tiagho Arsene Nelson

Élève ingénieur en travaux informatique option Génie Logiciel

**Sous l’encadrement :**

Stage effectué du 27 Juin au 31 Septembre 2021 en vue de l’obtention du Diplôme de Technicien Supérieur (DTS)

# DÉDICACE

**À ma Grande Mère**

# REMERCIEMENTS

# AVANT-PROPOS

# SOMMAIRE

DÉDICACE i

REMERCIEMENTS ii

AVANT-PROPOS iii

SOMMAIRE iv

RÉSUME v

ABSTRACT vi

LISTE DES ABREVIATIONS vii

LISTE DES FIGURES viii

LISTE DES TABLEAUX ix

INTRODUCTION GÉNÉRALE 1

PARTIE 1 : RAPPORTE D’INSERTION 2

CHAPITRE 1 : PRESENTATION DE L’ENTREPRISE 3

CHAPITRE 2 : DÉROULEMENT DU STAGE 7

PARTIE 2 : CAHIER DES CHARGES 9

CHAPITRE 1 : DOSSIER D’ÉTUDE 10

CHAPITRE 2 : DOSSIER CONCEPTION 17

CHAPITRE III : DOSSIER DE RÉALISATION 33

PARTIE III – BILAN 48

CHAPITRE 1 : PRÉSENTATION DE L’APPLICATION 49

CHAPITRE II – APPORTS 55

CONCLUSION GÉNÉRALE 56

WEBOGRAPHIE 56

TABLE DE MATIÈRES 56

ANNEXES 56

# RÉSUME

“Le temps est de l’argent” on le dit souvent bien, mais pour la microfinance ce n’est pas juste une métaphore. Car les décisions sont prises tout le temps et si on n’a pas la bonne information cela peut-être enregistrer comme une perte pour l’entreprise.

Pour aider ceux-ci tout gagner en temps le system propos une méthode de travail asynchrone car les ont devras plus attendre que les collectrices renter pour savoir combine on a en caisse. Ou encore attendre les jour ouvrable pour connaitre son solde pour les clients. Le système sera disponible 24H/24. Pour servir tant les clients que les employer.

# ABSTRACT

# LISTE DES ABREVIATIONS

# LISTE DES FIGURES

Figure 0 : schéma de la méthode 2TUP 18

Figure 01 : cas d’utilisation du client 22

Figure 02 : cas d’utilisation du collecteur 22

Figure 03 : cas d’utilisation de l’administrateur 23

Figure 04 : cas d’utilisation Gobal 24

Figure 05 : diagramme de séquence d’une collecte 29

Figure 06 : diagramme de séquence de la création d’un utilisateur par l’administrateur 30

Figure 07 : diagramme de séquence d’authentification 31

Figure 08 : diagramme de Classe 32

Figure 09 : logo XML 34

Figure 10 : logo Java 35

Figure 11 : logo Firebase 35

Figure 12 : logo React JS 36

Figure 13 : logo JavaScript 36

Figure 14 : logo HTML 37

Figure 15 : logo Sass 37

Figure 16 : logo GitHub 38

Figure 17 : logo node js 39

Figure 18 : logo PhotoShop 39

Figure 19 : logo Microsoft Word 40

Figure 20 : logo Microsoft PowerPoint 40

Figure 21 : représentation de la collection des utilisateur 41

Figure 22 : représentation de la collection des Comptes 42

Figure 23 : représentation de la collection des Transaction 43

Figure 24 : règle de création d’un document 43

Figure 25 : Diagramme de déploiement 47

# LISTE DES TABLEAUX

Tableau 0 : fiche d’identification de l’entreprise 3

Tableau 01 : équipe de travaille 13

Tableau 02 : ressource matérielle 13

Tableau 03 : ressources humaines 14

Tableau 04 : Total 14

Tableau 05 : planifications de taches 15

Tableau 06 : Contrainte du projet 15

Tableau 07 : formalisme digramme de cas d’utilisations 20

Tableau 08 : liste des cas d’utilisations 21

Tableau 09 : cas d’utilisation du changement de statut d’un utilisateur 24

Tableau 10 : cas d’utilisation de création d’un utilisateur 25

Tableau 11 : cas d’utilisation d’authentification d’un utilisateur 25

Tableau 12 : cas d’utilisation effectuer une collecte d’un utilisateur 26

Tableau 13 : cas d’utilisation de consulter le solde d’un utilisateur 26

Tableau 14 : liste des attributs 27

Tableau 15 : formalisme du diagramme de séquences 28

Tableau 16 : les cardinalités 32

Tableau 17 : Matériels de base 33

# INTRODUCTION GÉNÉRALE

Pour lier la théorie à la pratique, nous avons effectué un stage de quatre (04) mois, qui as débuté le 13 juin 2022 et a pris fin le 30 Septembre 2022. Le stage a été effectuer dans la microfinance SARKI MONEY CONTRIBUTION.

SARKI MONEY CONTRIBUTION est une structure offrant des services liés aux transactions financières tels que les comptes d’épargne, de crédit, courant, collecte journalière, retrait mobile money. SARKI MONEY CONTRIBUTION possède plusieurs centres dans plusieurs villes du Cameroun, celui où nous avons effectué notre stage est situé à Galim dans le département des Bamboutos. Notre stage était dans le cadre de technicien en conception de système informatique.

Etant présent pour étudier et voir dans quelle mesure améliorer les systèmes de gestion en place, dès les premières semaines nous avons remarqué une lenteur et une efficacité pas du tout optimal. Et pour palier à ces diffèrent problème nous avons choisis comme thème de Project : **mise en place d’une plateforme web et mobile de gestion des comptes dans une microfinance.**

Pour mieux exprimer le déroulement du stage, nous présenterons d’abord la phase d’insertion, le dossier de conception, le dossier de réalisation.

# PARTIE 1 : RAPPORT D’INSERTION

# INTRODUCTION

Arrive en entreprise le 13 Juin 2022, avons été accueillie par la directrice du centre Roukayatou Lady. En qualité de stagiaire. Dans les lignes qui suivent nous allons, d’abord présenté la structure, et ensuite le déroulement du stage.

# CHAPITRE 1 : PRESENTATION DE L’ENTREPRISE

## PRESENTATION GENERALE DE LA STRUCTURE

### I.1) Historique

Créée en septembre 2016 SARKI MONEY CONTRIBUTION, est une ramification de SARKI GROUP qui est constitué de plusieurs autres branches à savoir la logistique, la sécurité et le service immobilier. Notre sage c’est dérouler dans la microfinance. Sa majesté Inousa Djibril est le fondateur du Groupe. Son but premier est de montre au jeune l’importance de l’entreprenariat jeune.

Tableau 0 : fiche d’identification de l’entreprise

|  |  |
| --- | --- |
| Raison sociale | SARKI money contribution (SMC) |
| Logo |  |
| Contact |  |
| Date de création | **No table of figures entries found.**12 Septembre 2016 |
| Siege sociale | Douala Cameroun |
| Responsable | ROUKAYATOU LADY |
| Capital social | 556 000 000 FCFA |
| Page Facebook | SARKI Group |
| Slogan | « Servir bien pour, servir toute la vie » |

### I.2) Situation géographique

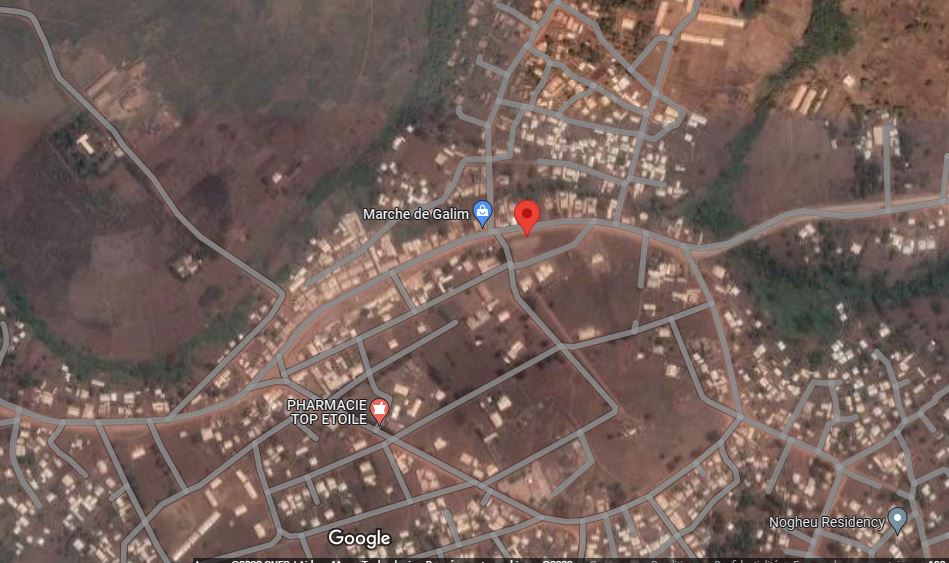


Figure 0: Situation géographique

## MISSIONS ET ORGANES DE GESTION

### II.1) Services

Les services offerts par celle-ci son multiple à savoir :

* Ouverture des comptes courant : c’est compte réserver pour les transactions financières de façon récurrente.
* Ouverture des comptes Épargne : est un type de compte qui peux générer des taux d’intérêt annuelle pouvant aller jusqu’à 6%.
* Le compte de collecte : qui permet au client de faire des collectes journalières
* Le crédit de construction
* Le crédit pour ouverture d’un business

Pour mieux servir ces clients l’entreprise est ouvert de lundi à samedi de 8h00 à 15H00.

### II.2) Missions

Lors de sa création la microfinance c’est donné comme mission :

* Accompagner les jeunes dans la réalisation de leur rêve.
* Le soutien financier aux populations.
* Pousser les jeunes à entreprenariat.

### II.3) Organes de gestion

Pour le bon fonctionnement de l’entreprise, elle possède comme organe de gestion :

* Le directeur général : qui est charger de cordonne les opérations des différentes agences
* Le chef d’agence
* Une caissière
* Une collectrice
* Service clients

Collectrice

Directeur Général

Chef d’agence

Caissière

Service clients

# CHAPITRE 2 : DÉROULEMENT DU STAGE

## ACCUEIL

Lundi le 14 Juin 2022 marque le début de notre stage académique à la microfinance SARKI MONEY CONTRIBUTION, devant se poursuivre jusqu’au 30 Septembre 2021. Les premières semaines dite période d’insertion, ont été marquée par plusieurs étapes : Tout d’abord la prise en contact avec les personnels qui après les questions de civilité d’usage nous ont remis une fiche de renseignement à remplir. Il était question ici de décrire notre identité et notre niveau d’étude. Après avoir pris connaissances du règlement intérieur pour stagiaires. Une fois la prise de contact établie, nous avons rencontré Mme R. Lady qui après un bref entretien nous a exhorté de nous mettre immédiatement au travail.

## TÂCHES EFFECTUÉES

Dès le premier jour, nous avons effectué quelques taches dans la structure parmi lesquelles on peut citer :

* + Restauration d’un système d’exploitation ;
  + Résolution des imprimantes ;
  + Installation de Microsoft office 2010 sur plusieurs postes ;
  + Installation de l’anti-virus Avast ;
  + Test et rangement de tous les appareils ;
  + Installation du système d’exploitation ;
  + Vérification des machines ;
  + Aménagement de mon espace de travail

## ÉVALUATION

### III.1) Difficultés

Le plus difficile l’or de notre stage était les soucis rencontrés avec les coupures récurrentes d’électricité.

### III.2) Bilan de l’expérience

Tout d’abord, il est important de noter que notre stage s’est déroulé dans des bonnes conditions. Au début, l’adaptation et l’insertion au sein de l’entreprise n’a pas été facile mais grâce à nos efforts, nous avons pu nous intégrer dans le nouvel environnement et cela nous a permis de comprendre à peu près comment fonctionne les microfinances.

Pour mener à bien notre stage, il a fallu que nous développions certaines attitudes notamment

* La sociabilité pour pouvoir créer et entretenir une bonne relation de travail avec les camarades et les employés de la structure.
* Le développement de l’écoute (surtout être beaucoup attentif) pour ne pas laisser s’échapper un détail qui pourrait être utile ou important pour notre stage.
* L’humilité et la facilité de porter assistance au mieux de nos compétences à tous les utilisateurs dans le besoin.

# PARTIE 2 : CAHIER DES CHARGES

# INTRODUCTION

Le cahier des charges (CDC) est un document contractuel à respecter lors d'un projet. Le cahier des charges permet au maître d'ouvrage de faire savoir au maître d'œuvre ce qu'il attend de lui lors de la réalisation du projet, entraînant des pénalités en cas de non-respect. Ainsi notre cahier de charge est constitué d’un dossier d’étude, le dossier de conception et pour finir le dossier de réalisation.

# CHAPITRE 1 : DOSSIER D’ÉTUDE

## CONTEXTE ET JUSTIFICATION DU PROJET

Dans la vie au quotidien les microfinances sont amenées à servir les populations. Et pour évite de faire faillite la microfinance met en place des systèmes qui vont la permettre de surveiller les comptes et les différentes actions mené sur ce dernier. Car avec les microfinances le temps c’est de l’argent. C’est dans le souci d’optimiser la surveillance que notre projet : **mise en place d’une plateforme web et mobile de gestion des comptes dans une microfinance**, prend tout son sens.

## ÉTUDE DE L’EXISTANT

Le système existant présente des forces et des faiblesses d’où une présentation se fera en deux parties.

### II.1) Présentation

Le système existant dans l’entreprise est un système principale base sur des enregistrements synchrone (les données venant de la fiche de la collectrice sont enregistré dans une feuille de calcul Excel). Dans la machine de la caisse est enregistré toutes les donnes sur les clients et leurs différentes transactions. Pour faciliter le rangement, les feuilles sont classées par client, par transactions, par compte.

### II.2) Point fort

* La présence d’une traçabilité des donner des différentes transactions.
* La collaboration et la présence de la confiance entre les différents acteurs : client, collecteur, administration

### II.3) Point faible

* Absence d’une base de données relationnelle : ici toutes les données sont stockées sur des feuilles de calcul Excel.
* Manque des états en temps réels : les donnes sont collectées qu’en fin de journée au retour des collectrices.
* Pertes récurrentes des donner : car souvent on ne retrouve pas une fiche ou carnet.
* Falsification des fiches de collecte : car on remarque sur les fiches la présence des marques du correcteur qui prouve ce qui a été initialement écrit a été modifié.
* Couteux du fais de l’impression permanente des fiches pour les collecte et l’impression des carnets de collecte
* Difficulté pour l’administration de prendre des décisions en temps réels : car elle doit attendre le retour des collectrices.

D’où nous pouvons affirmer que nous avons un système fonctionnel avec beaucoup d’inconvénient. Ainsi notre système viendra résoudre tous les points faibles de ce système déjà existant et rendre la micro finance plus performante.

### II.4) Proposition de solution

Pour pallier à ce problème rencontre par les clients à avoir le solde de leur compte n’importe où, et a administration d’avoir les informations en temps réelle. Notre système sera d’une grande aide pour résoudre tous les c’est problème et améliorer les services déjà offert par le précèdent. Ainsi avec son téléphone mobile Android. Le client pourra voir toutes les transactions effectuer dans son compte n’importe où. Et l’administration réservera les transactions en temps réelle car tout sera stoker sur un server hébergé par Google.

## LES OBJECTIFS DU PROJET

### III.1) Objectif pour l’administration

Le projet a pour objectif principale d’optimiser la surveillance des comptes. Cette surveillance optimiser vas permettre à la direction de savoir les différentes opérations sur les diffèrent compte en temps réel. Car Grace à ces informations l’administration pourras prendre des décisions à temps, s’il faut accorder le crédit à un client.

### III.2) Objectif pour le client

Le projet va aider les clients car ceux-ci pourront depuis leur téléphone voir en temps réel l’historique des transactions faite sur ces comptes.

### III.3) Objectif pour les collecteurs

Pour les collecteurs le projet va leur aider à mieux effectuer les collectes, dans la mesure où ils pourront depuis leur téléphone enregistrer les collectes des clients sans stresse. Et en plus elle pourra voir les statistiques sur ces activités.

## EXPRESSIONS DES BESOINS DE L’UTILISATEUR

### IV.1) Besoin fonctionnel du projet

Les besoins fonctionnels sont les besoins qui aboutissent à une fonctionnalité dans le système développé, elle représente une action ou un ensemble d’actions à exécuter pour aboutir à un résultat donné. Les besoins fonctionnels du système sont groupés en catégories comme suit :

### IV.2) Au niveau du client

* S’inscrire
* Se connecter
* Se déconnecter
* Voir le listing des transactions
* Voir le solde de son (ses) compte(s)

### IV.3) Au niveau du collecteur

* Voir le listing des transactions
* Pourvoir ouvrir un compte pour un client

### IV.4) Au niveau de l’administrateur

* Créer un utilisateur
* Changer les statuts des diffèrent utilisateur : il peut transformer un utilisateur standard en collecteur.
* Ouvrir un compte
* Voir toutes les transactions du système
* Vérifier le solde des différents comptes
* Imprimer les fiches de solde
* Imprimer le reçu de transactions

### IV.5) Besoin non fonctionnel du projet

Ce sont des besoins qui devront rendre le système plus accessible et plus convivial. Permettant ainsi l’utilisateur d’être satisfait du système.

### IV.6) Fiabilité

L’application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.

### IV.7) Ergonomie et bonne interface

L’application doit être adaptée à l’utilisateur sans qu’il ne fournisse aucun effort (utilisation claire et facile) de point de vue navigation entre les différentes pages.

### IV.8) Sécurité

Notre solution doit respecter surtout la confidentialité des données personnelles des utilisateurs qui reste l’une des contraintes les plus importantes dans notre système.

### IV.9) Aptitude à la maintenance et la réutilisation

### IV.10) L’intégrité

Les donnes enregistrer ne doivent pas être modifié par une tierce personne sans autorisation ni être perdus.

### IV.11) Disponibilité : 24h/24 et 7j/7 BESOINS

## ESTIMATION DU COÛT DU PROJET ET EQUIPE DE TRAVAIL

Pour la réalisation de ce projet, nous devons avoir à notre disposition de ressources matérielles, logiciels, humaines et également financières, que nous avons listés comme suite :

### V.1) Équipe de travail

Tableau 01 : équipe de travaille

|  |  |
| --- | --- |
| **Noms et Prénom** | **Fonction** |
| LONTCHI Loïc Steve | Modélisation, Conception, et Réalisation |
| M. MELI TAMETANG Maxim | Encadreur académique et chargé de communication à l’IAI Cameroun Centre technologique d'excellence de Paul Biya |
| Mme ROUKAYATOU Lady | Chef de projet. Encadreur professionnel. Directrice de SARKI MONEY CONTRIBUTION |

### V.2) Resource matérielles

Tableau 02 : ressource matérielle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Matériels** | **Fonctions** | **Prix unitaire** | **Quantité** | **Prix totale (FCFA)** |
| **Ordinateur : i5 5e Gene 8Go Ram** | Équipement de développement | **250 000** | **03** | **750 000** |
| **Téléphone Android : 7.1, 1Go ram** | Équipement de test | **50 000** | **02** | **100 000** |
| Box wifi Orange | Accès à internet | **39 900** | **01** | **39 900** |
| Total | **889 900 F CFA** | | | |

### V.3) Ressources humaines

NB : La source de chaque prix est disponible et détaillée dans la webographie

Tableau 03 : ressources humaines

|  |  |  |
| --- | --- | --- |
| **Main d’œuvre** | **Nombres** | **Salaires** |
| **Analyste & concepteur** | 02 | 400 000 |
| **Designer** | 01 | 100 000 |
| **Programmeur** | 02 | 200 000 |
| **Tester** | 15 | 300 000 |
| **Total** | 15 | 1 000 000 |

### V.4) Grand total

Tableau 04 : Total

|  |  |
| --- | --- |
| **Types** | **Prix (F CFA)** |
| **Matérielles** | **889 900** |
| Humaines | **1 000 000** |
| Imprévues | **1 889 900 \* 10%** |
| Total de cout | **2 078 890** |

## PLANIFICATION DU PROJET

Comme tout projet de conception un planning des étapes dois être établir et connues. Pour notre projet les étapes sont les suivant.

L’étude préalable : Le résultat de cette phase est la détermination des objectifs à atteindre dans notre future application en partant de l’existant.

Conception : Il s’agit de détailler les spécifications des fonctions ainsi que la structure des données, et des contrôles et les interfaces.

Réalisation : Il s’agit de l’implémentation des différentes fonctionnalités du programmes et effectuer les tests unitaires.

Test et Validation : Il s ‘agit de tester notre plateforme en lui faisant passés des audits de validation qui consisterons à détecter tout comportement inapproprié (faille) du système et le corriger.

Rédaction du rapport : Description détaillée de notre travail.

Tableau 05 : planifications de taches

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Étape/ semaine** | **Juin** | | | | **Juillet** | | | | **Aout** | | | | **Septembre** | | | |
|  | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **Étude préalable** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Conception** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Réalisation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Test et validation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Rédaction du rapport** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## LES CONTRAINTES DU PROJET

Une contrainte est un élément qui limite un projet de par son potentiel par rapport à ses objectifs, Pour notre projet, nous devrons faire face à trois contraintes qui sont les suivantes ; Contrainte de temps, contrainte de coûts et contrainte de ressources.

Tableau 06 : Contrainte du projet

|  |  |
| --- | --- |
| **Contraintes** | **Définitions** |
| Contraintes de temps | Limités sur le temps de début et de fin de chaque tâche dans le chemin critique d'un projet, qui est la séquence de tâches qui ne peuvent être retardées sans retarder l'ensemble du projet. Les retards dans les étapes critiques d'un projet sont habituellement les principaux éléments qui affectent le temps d'achèvement du projet. Dans notre cas nous faisons face à une durée de temps très impartis ce a causse des quatre mois de stage. |
| Contraintes de coûts | Les contraintes de coûts comprennent le [budget du projet](https://asana.com/fr/resources/project-budget) dans son ensemble et tout élément de valeur financière nécessaire à votre projet. |
| Contraintes de ressources | Les ressources sont étroitement liées aux contraintes de coûts de votre projet, car ces dernières représentent un certain coût. Une mauvaise [allocation des ressources](https://asana.com/fr/resources/resource-allocation) peut entraîner une baisse de qualité du projet, une augmentation du budget et des retards dans le calendrier. |

1. **LES LIVRABLES**

A la fin du délai fixé pour le développement de cette solution, les éléments qui constituent les livrables sont :

* Le code source de l’application
* Le dossier de conception
* Le dossier de réalisation
* Un manuel d’utilisation

**Conclusion**

L’élaboration de notre cahier de charges nous a ainsi permis d’avoir une idée précise des exigences nécessaires à la réalisation de notre plateforme. Nous allons aborder avec beaucoup de lucidité la prochaine partie qui est le dossier d’analyse et de conception.

# CHAPITRE 2 : DOSSIER CONCEPTION

La démarche de conception est une étape fondamentale dans le processus de développement, puisqu’elle fait correspondre la vision applicative (le modèle d’analyse) à la vision technique (l’environnement de développement et d’exécution).

Cette partie vise à illustrer la phase de conception des diagrammes UML. Nous commençons par établir les diagrammes de cas d’utilisation, en suite les diagrammes de séquences, après nous élaborons le diagramme de classes.

## PRÉSENTATION DE LA MÉTHODE 2TUP

Devant le nombre considérable de méthodes disponible, le choix devient difficile. Beaucoup de questions peuvent être posées à un chef de projet lors du lancement d'un projet : comment vais-je organiser l'équipe de développement ? Comment faire participer le client au développement afin de capter les besoins de celui-ci ? Pour y répondre, notre choix s'est porté sur la méthode 2TUP car c'est une méthode est guidée par les besoins de l'utilisateur. 2TUP est un processus unifié qui a pour but d'apporter une réponse aux contraintes de changement fonctionnelle et techniques qui s'imposent aux systèmes d'information. Il propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels. Il part du constat que toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement suivant un axe fonctionnel et un axe technique. Il distingue ainsi deux branches (fonctionnelle et technique) dont les résultats sont fusionnés pour réaliser la solution finale. Il faut noter que le processus 2TUP commence d'abord par une étude préliminaire. Dans cette dernière, il s'agit d'identifier les acteurs qui vont interagir avec le système, les messages qu’échappent les acteurs et le système, puis à produire un cahier de charges et enfin à modéliser le contexte. A l’issue des évolutions du model fonctionnel et de l’architecture technique, la réalisation du système consiste à fusionner le résultat des deux branches à savoir la branche fonctionnelle et la branche technique. Cette fusion conduit à l’obtention d’un processus de développement en forme Y, comme l’illustre le schéma suivant :

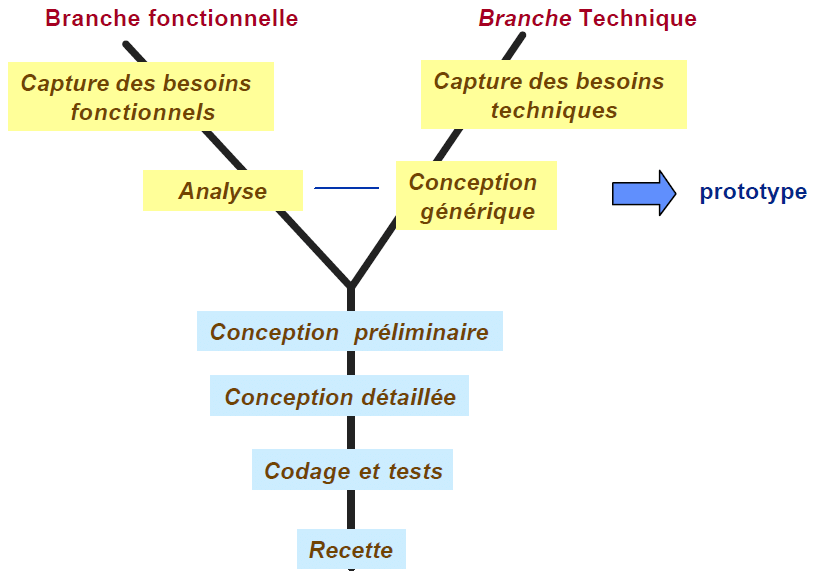


Figure 0 : schéma de la méthode 2TUP

## CONCEPTION

Le Modèle conceptuel de données est une représentation statique du système d’information. Il a comme objectif de constituer une représentation claire et cohérente des données manipulées dans le système d’information.

Dans le souci de normaliser notre conception nous devons choisir une méthode. Nous allons ainsi présenter la méthode utiliser : UML (Unified Modelling Language).

## PRÉSENTATION DE UML

UML (Unified Modelling Language) est un langage formel et normalisé en termes de modélisation objet. Son indépendance par rapport aux langages de programmation, aux domaines de l’application et aux processus, son caractère polyvalent et sa souplesse ont fait de lui un langage universel. En plus UML est essentiellement un support de communication, qui facilite la représentation et la compréhension de solution objet. Sa notation graphique permet d’exprimer visuellement une solution objet, ce qui facilite la comparaison et l’évaluation des solutions. L’aspect de sa notation, limite l’ambigüité et les incompréhensions. UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues.

Une vue est constituée d'un ou plusieurs diagrammes. On distingue deux types de vues :

* + - La vue statique, permettant de représenter le système physiquement :
* Diagrammes de classes : Représentent des collections d'éléments de modélisation statiques (classes, paquetages...), qui montrent la structure d'un modèle.
* Diagrammes d'objets : Ces diagrammes montrent des objets (instances classes dans un état particulier) et des liens (relations sémantiques) entre objets.
* Diagrammes de cas d'utilisation : Identifient les utilisateurs du système (acteurs) et leurs interactions avec le système.
* Diagrammes de composants : Permettent de décrire l'architecture physique statique d'une application en termes de modules : fichiers sources, librairies exécutables, etc.
* Diagrammes de déploiement : Montrent la disposition physique du matériel qui compose le système et la répartition des composants sur ce matériel.
  + - La vue dynamique, montrant le fonctionnement du système :
* Diagrammes de collaboration : Montrent des interactions entre objets (instances de classes et acteurs).
* Diagrammes de séquence : Permettent de représenter des collaborations eu objets selon un point de vue temporel, on y met l'accent sur la chronologie (envois de messages).
* Diagrammes d'états-transitions : Permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.
* Diagrammes d'activités : (une variante des diagrammes d'états-transitions) servent à représenter graphiquement le comportement d'une méthode ou déroulement d'un cas d'utilisation.

Les informations citez dans le paragraphe ci-dessus, ont été collecté sur le site officiel de la méthode UML.

La conception de notre système a été élaborée en suivant la démarche suivante :

* L'élaboration des diagrammes de cas d'utilisation. Cette étape a été réalisée suite à la spécification fonctionnelle de l’application.
* Recensement des classes candidates et élaboration du diagramme des classes.
* Dresser les diagrammes de collaboration et de séquences pour mettre en évidence interactions entre les différents objets du système.
* Élaborer le diagramme d'états-transitions pour montrer les différents états d'interface.

### III.1) Diagramme fonctionnel

#### III.1.1) Diagramme des cas d’utilisation

Le diagramme de cas d’utilisation (Use Case Diagram) constitue la première étape de l’analyse UML en :

* Modélisant les besoins des utilisateurs.
* Identifiant les grandes fonctionnalités et les limites du système.
* Représentant les interactions entre le système et ses utilisateurs.
* Composant de cas d’utilisation

Tableau 07 : formalisme digramme de cas d’utilisations

|  |  |  |
| --- | --- | --- |
| **Composant** | **Description** | **Représentation graphique** |
| Acteur | Un acteur est une entité extérieure au système modélisé, et qui interagit directement avec lui. Il en existe 2 types d’acteurs : Primaire et secondaire. |  |
| Cas d’utilisation | Un use case (cas d’utilisation) représente une fonctionnalité du système. Cette fonctionnalité est définie par une action déclenchant un ou plusieurs déroulements possibles. |  |
| Association | Les acteurs et les cas d'utilisation sont reliés par des associations. Ainsi que les acteurs du système entre eux. |  |

Notre système est constitué de trois groupes d’utilisateurs qui interagissent avec ce dernier en réalisant un ou plusieurs cas d’utilisations. Comme acteur, nous avons :

* Le client : C’est un utilisateur qui visualise ces diffèrent compte
* Le collecteur : celui-ci utilise le système pour faire des collectes
* Administrateur : Celui-ci a le contrôle du système entier, c’est-à-dire qu’il a accès à l’intégralité du système sauf sur restriction.

Les interactions entre les acteurs et le système seront représentés grâce au diagramme de cas d’utilisation du langage de modélisation UML 2.0. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

Tableau 08 : liste des cas d’utilisations

|  |  |
| --- | --- |
| **CAS D’UTILISATION** | **ACTEURS IMPLIQUES** |
| Changer le statut des utilisateurs | Administrateur |
| Consulter le solde de son compte | Client |
| Créer un utilisateur | Administrateur |
| Effectuer une collecte | Collecteur, administrateur |
| Faire un dépôt | Administrateur |
| Faire un listing de ces transactions | Client, collecteur |
| Faire un retrait | Administrateur |
| Ouvrir tout type de compte | Administrateur |
| Ouvrir un compte collecte | Collecteur, administrateur |
| Voir ces statistiques | Client, collecteur, administrateur |

#### III.1.2) Diagramme de cas d’utilisation du client

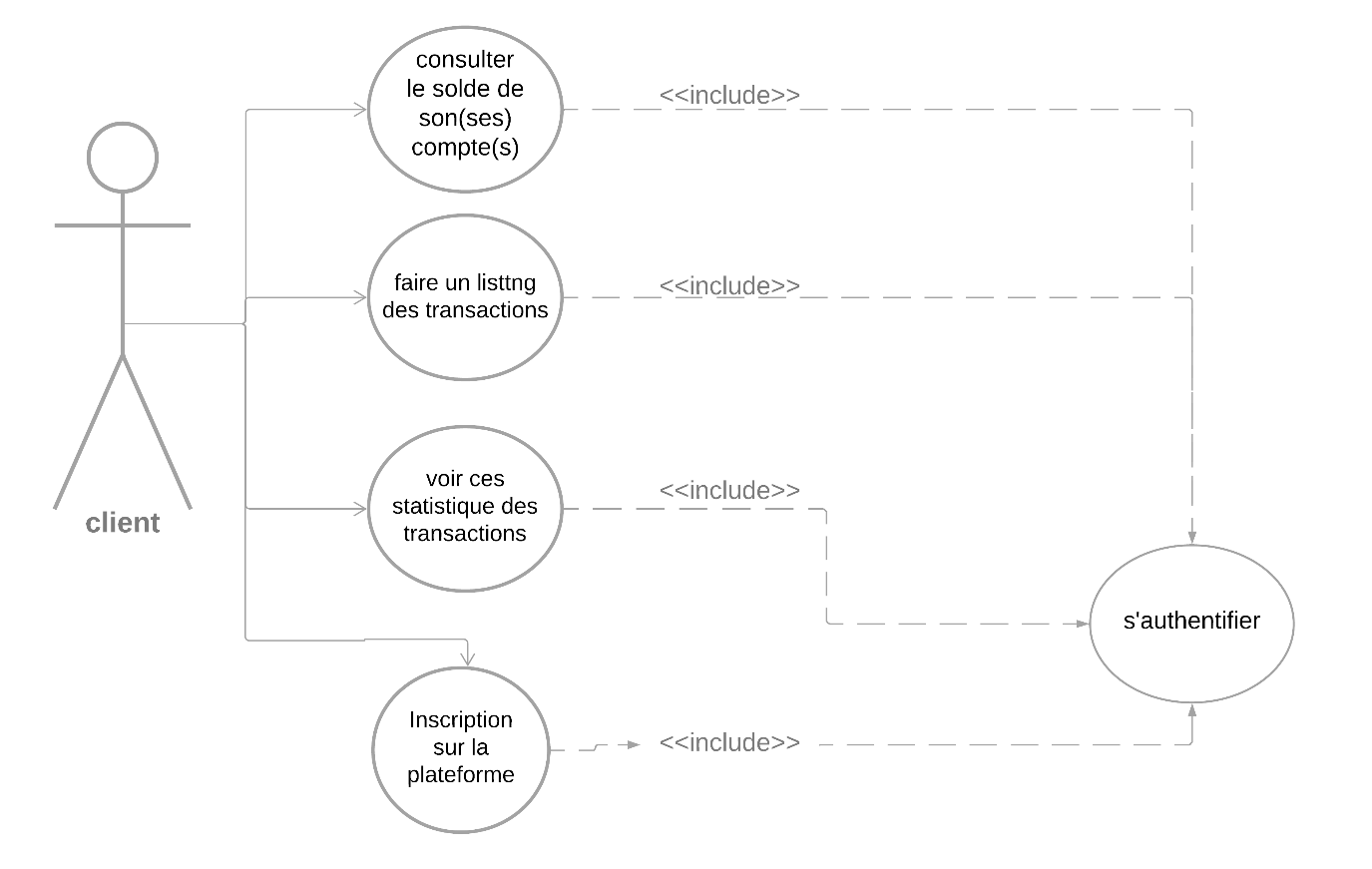


Figure 01 : cas d’utilisation du client

#### III.1.3) Diagramme de cas d’utilisation du collecteur

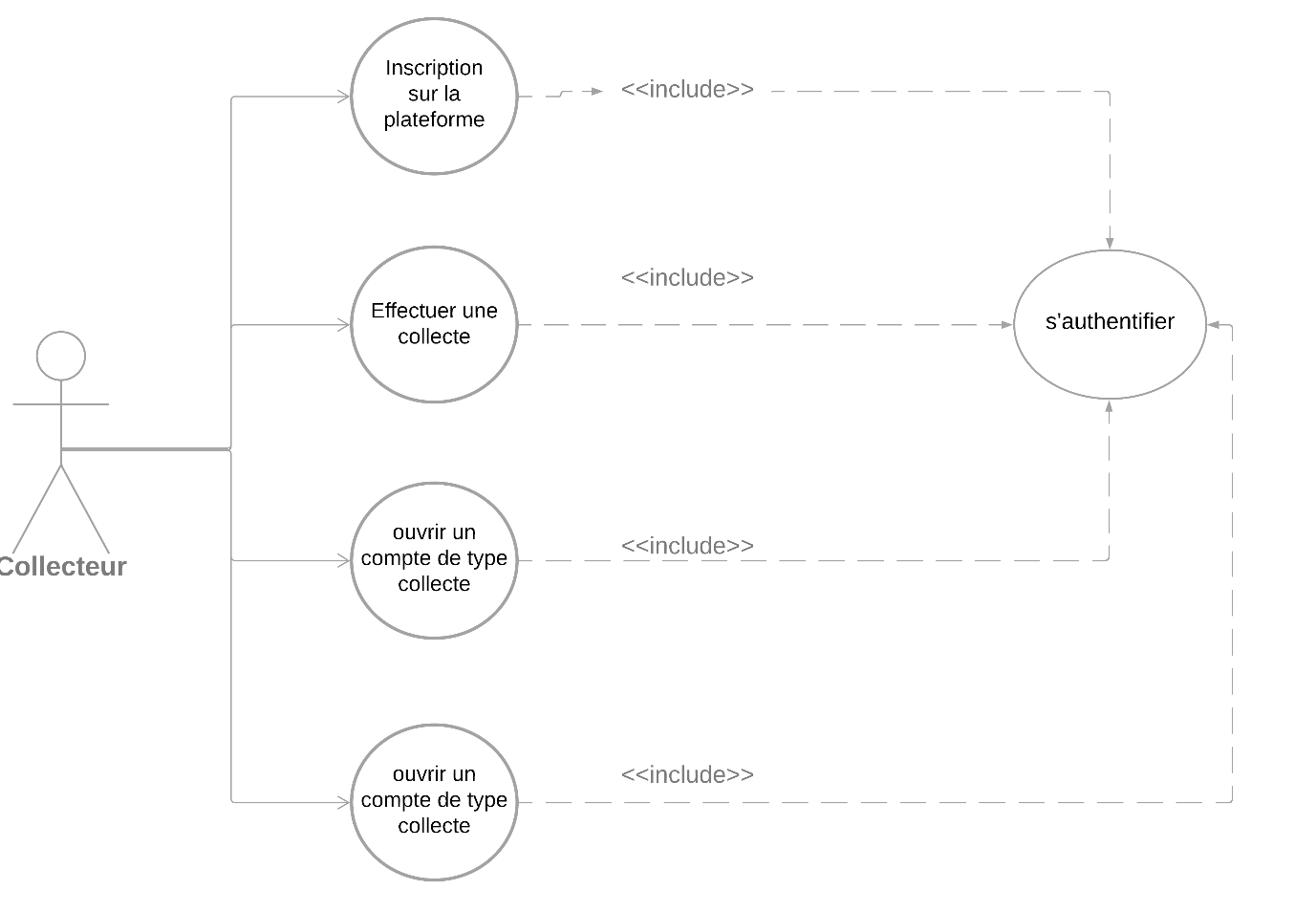


Figure 02 : cas d’utilisation du collecteur

#### III.1.4) Diagramme de cas d’utilisation de l’administrateur

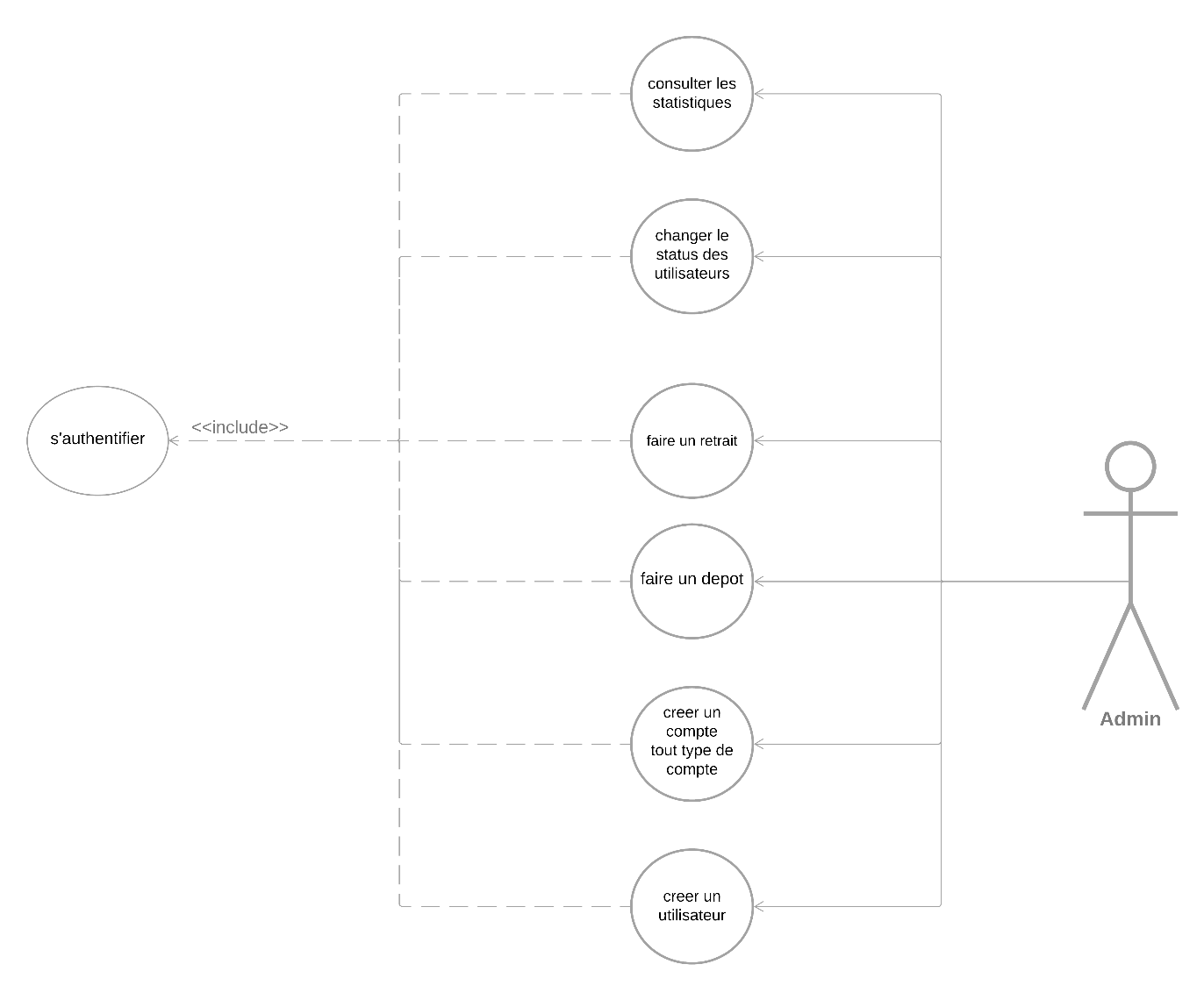


Figure 03 : cas d’utilisation de l’administrateur

#### III.1.5) Diagramme de cas d’utilisation global

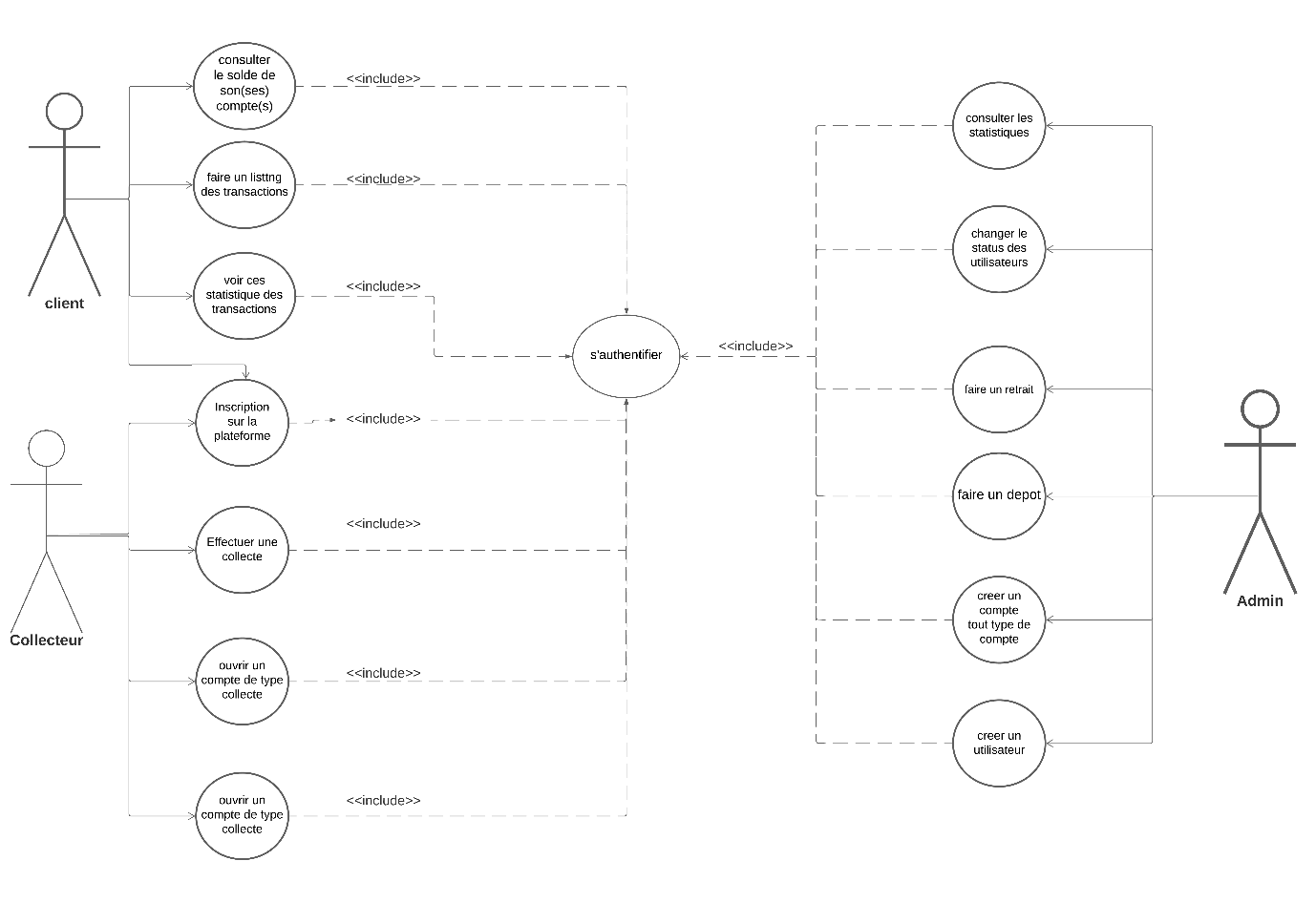


Figure 04 : cas d’utilisation Global

#### III.1.6) Description des cas d’utilisation

Dans le but de mieux comprendre notre système et les interactions avec les utilisateurs, dans cette partie nous allons détailler les scenarios de principaux cas d’utilisation.

Tableau 09 : cas d’utilisation du changement de statut d’un utilisateur

|  |
| --- |
| CU1 : Changer le statut des utilisateurs |
| Résumé : permet à l’administrateur de faire des mises à jour sur le statut des utilisateurs |
| Acteurs : administrateur |
| Scénario nominal |
| DESCRIPTION DU SCENARIO NOMINAL  « Début »  01 : authentification  02 : choisir l’utilisateur  03 : définir un statut  04 : confirmer  « Fin » |
| Scenario alternative |
| L’utilisateur n’est pas autorisé à faire des changements de statut : mise à jour non autoriser  L’utilisateur sélection possède déjà un compte de n’importe qu’elle type : échec de la mise à jour |

Tableau 10 : cas d’utilisation de création d’un utilisateur

|  |
| --- |
| CU2 : Créer un utilisateur |
| Résumé : crée un utilisateur si celui-ci ne possède pas de mobile Android |
| Acteur : administration |
| Scénario nominal |
| DESCRIPTION DU SCENARIO NOMINAL  « Début »  01 : authentification  02 : saisir les informations de l’utilisateur  03 : confirmer la création  « Fin » |
| Scénario alternative |
| DESCRIPTION DU SCENARIO ALTERNATIF  Les informations entrées ne sont pas valides : veuillez vérifier les informations entre est le message d’erreur retourné. |

Tableau 11 : cas d’utilisation d’authentification d’un utilisateur

|  |
| --- |
| CU3 : authentification |
| Résumé : permettre au diffèrent acteur d’avoir accès au système. |
| Acteur : administration, client, collecteur |
| Scénario nominal |
| DESCRIPTION DU SCENARIO NOMINAL  « Début »  01 : saisir des crédenciale (information d’authentification)  02 : cliquer sur connecter  « Fin » |
| Scénario alternative |
| DESCRIPTION DU SCENARIO ALTERNATIF  Les informations entrées ne sont pas valides : veuillez vérifier les informations entre. Et a 05 tentative le compte est désactiver et devra se réfère à l’administrateur pour réactiver le compte. |

Tableau 12 : cas d’utilisation effectuer une collecte d’un utilisateur

|  |
| --- |
| CU4 : Effectuer une collecte |
| Résumé : permettre d’approvisionner le compte de type collecte |
| Acteur : collecteur, administrateur |
| Scénario nominal |
| DESCRIPTION DU SCENARIO NOMINAL  « Début »  01 : saisir le numéro du compte et le montant  02 : cliquer sur collecter  03 : confirmer la collecte  « Fin » |
| Scénario alternative |
| DESCRIPTION DU SCENARIO ALTERNATIF   * Le montant saisir est inférieur au montant minimal : erreur sur le montant * La collecte n’est pas autorisée : vous ne pouvez effectuer les collectes que sur les comptes qui vous été assigner. |

Tableau 13 : cas d’utilisation de consulter le solde d’un utilisateur

|  |
| --- |
| CU5 : consulter son solde |
| Résumé : permettre d’approvisionner le compte de type collecte |
| Acteur : client |
| Scénario nominal |
| DESCRIPTION DU SCENARIO NOMINAL  « Début »  01 : s’authentification  « Fin » |
| Scénario alternative |
| DESCRIPTION DU SCENARIO ALTERNATIF  Erreur sur les crédenciales saisit : email ou mot de passe incorrect.  Problème de réseau alors les informations ne seront pas mises à jour. |

### III.2) Modélisation conceptuelle des données

La modélisation conceptuelle des données permet de dégager l'ensemble des données manipulées en vue d'élaborer le diagramme de classes.

En effet, ce dernier donne une vue statique du système. Il décrit les types et les objets du système. Il s’agit donc d’une représentation des données du champ de l’étude ainsi que le lien sémantique entre ces données, facilement compréhensible, permettant de décrire le système d’information à l’aide des concepts proposés par le modèle UML.

#### III.2.1) Dictionnaire des données

Le tableau ci-dessous représente la liste des attributs composants toutes les classes formants notre système ainsi que leur description, leur taille et leur type.

Tableau 14 : liste des attributs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numéro** | **Attribut** | **Libelle** | **Type** | **Taille** |
| 01 | Id\_user | Identifiant utilisateur | Timestamp | // |
| 02 | Id\_compte | Identifiant du compte | Timestamp | // |
| 03 | Id\_transaction | Identifiant d’une transaction | Timestamp | // |
| 04 | Nom\_prenom | Nom et prénoms de l’utilisateur | Chaine | 50 |
| 05 | dateTransaction | Date de création de la transaction | Date | // |
| 06 | montant | Montant de la transaction | Réel | // |
| 07 | typeCompte | Type de compte | Chaine | 30 |
| 08 | Type | Type de transaction | Chaine | 20 |
| 09 | effectuer | Grade d’utilisateur ayant effectué la transaction | Chaine | 20 |
| 10 | libeleCompte | Libelle du compte | Chaine | 20 |
| 11 | Email | Email de l’utilisateur | Chaine | 30 |
| 12 | Location | Adresse du client | Chaine | 30 |
| 13 | Occupation | Profession de l’utilisateur | Chaine | 30 |
| 14 | numeroCNI | Numéro de la carte d’identité national | Chaine | 30 |
| 15 | Status | Statu de l’utilisateur | Chaine | 30 |
| 16 | Telephone | Numéro de téléphone de l’utilisateur | Entier | 10 |
| 17 | dteCreaction | Date de création du compte | Date |  |
| 18 | TypeCompte | Type de compte | Chaine | 20 |

#### III.2.2) Diagramme de séquence

Le diagramme des séquences permet de présenter une documentation des interactions à mettre en œuvre entre les classes pour réaliser un résultat. UML étant conçus pour la programmation orientée objet, les communications entre les classes sont connues comme des messages. Ce diagramme énumère les objets horizontalement et le temps verticalement. Chaque message est modélisé en fonction du temps.

Tableau 15 : formalisme du diagramme de séquences

|  |  |  |
| --- | --- | --- |
| **Composant** | **Description** | **Représentation graphique** |
| **Objet** | Les objets sont les instances de classe et sont rangés horizontalement. Acteur Personne qui interagit ou communique avec le système |  |
| **Ligne de vie** | La ligne de vie identifie l’existence de l’objet par rapport au temps |  |
| **Les messages** | Qui vont de l’acteur vers l’objet sont dits « synchrones » et ceux qui vont de l’objet vers l’acteur sont dits « asynchrones » |  |
| **Activation** | Indique quand l’objet effectue une action Message Indique les communications entre les objets. |  |

### III.3) Quelques diagrammes de séquences du système

#### III.3.1) Collecte d’un client

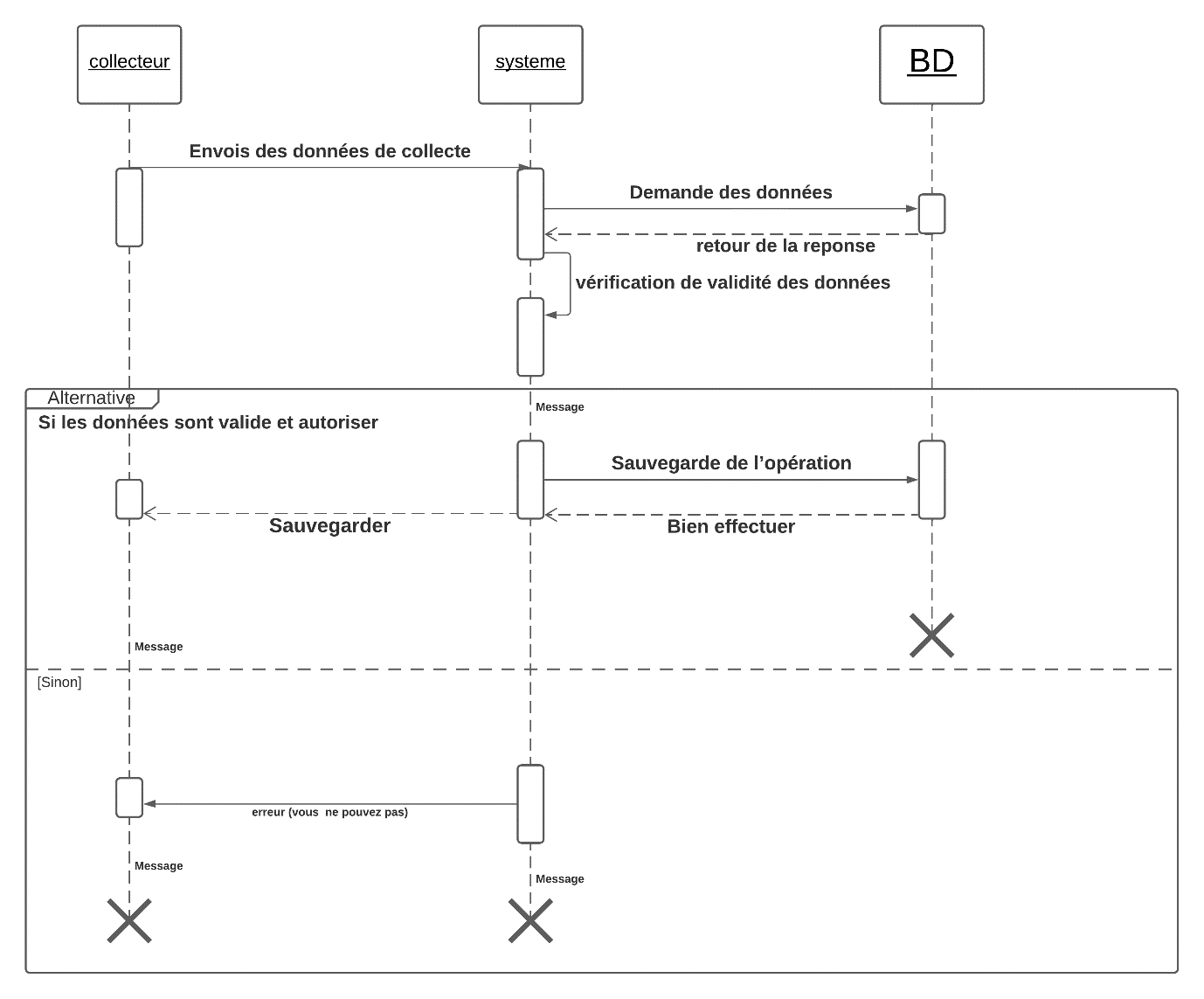


Figure 05 : diagramme de séquence d’une collecte

#### III.3.2) Création d’un utilisateur

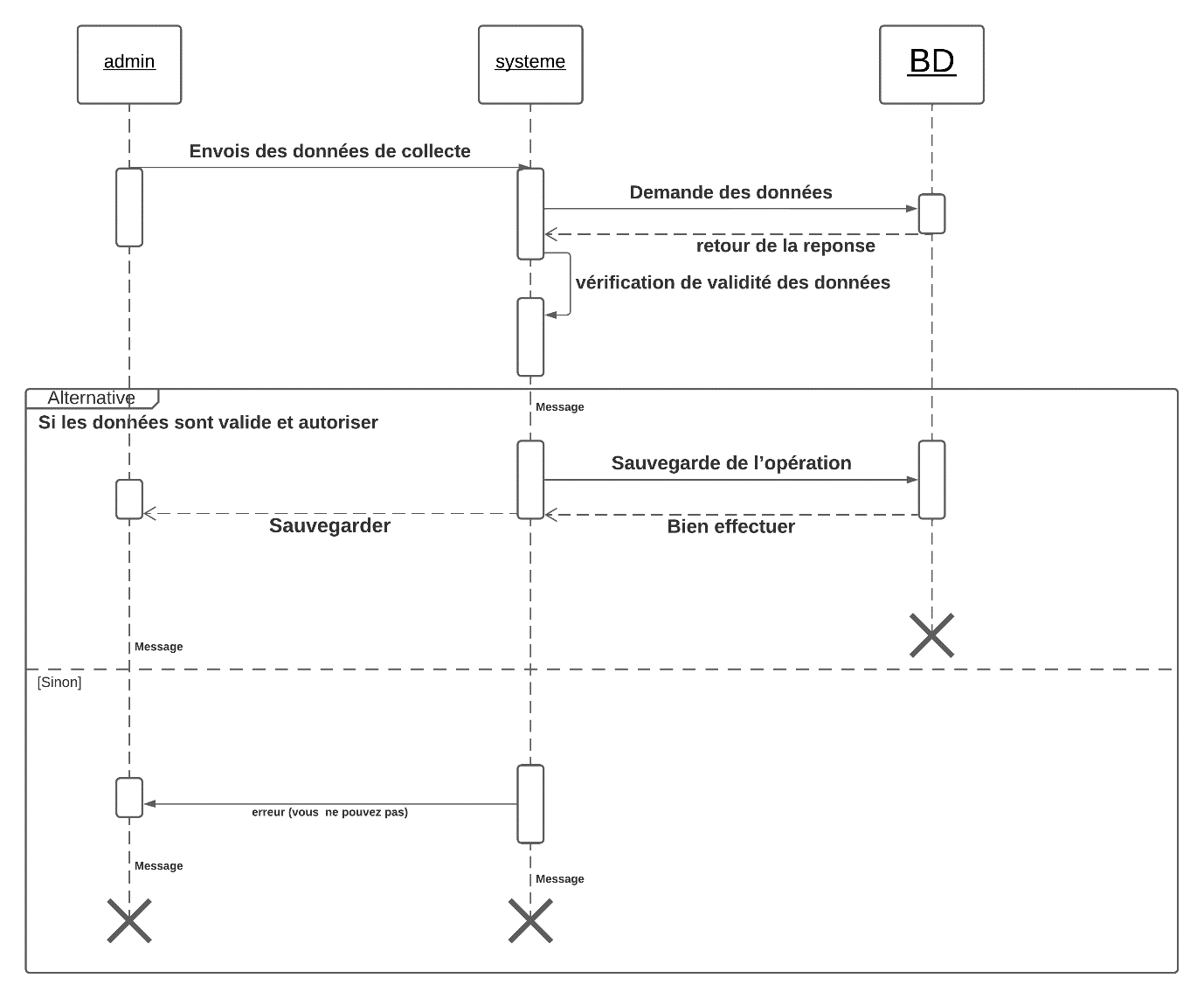


Figure 06 : diagramme de séquence de la création d’un utilisateur par l’administrateur

#### III.3.3) Authentification

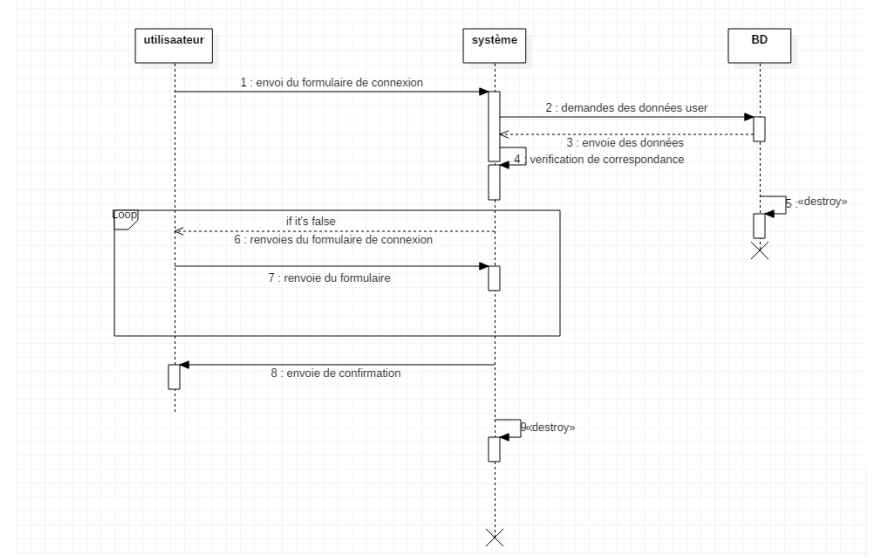


Figure 07 : diagramme de séquence d’authentification

### III.4) Représentation des classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation

Une classe est composée :

* Attributs : représentant des données dont les valeurs représentent l’état de l’objet.
* Méthode : il s’agit des opérations applicables aux objets.

Les associations sont des relations entre classes. Elles représentent une collaboration. Elles sont représentées par une ligne entre les classes.

Association simple : les associations simples sont des liaisons logiques entre entités.

Les cardinalités : précisent combien d’objets de classe considérée peuvent être liés à un objet de l’autre classe.

Tableau 16 : les cardinalités

|  |  |
| --- | --- |
| **Cardinalités** | **Désignation** |
| 1 / 1..1 | Un et un seul |
| 0..1 | Zéro ou un |
| N | Entier naturel |
| m..n | De m à n (deux entiers naturels) |
| 0..\* | De 0 à plusieurs |
| 1..\* | De 1 à plusieurs |

La figure ci-dessous est celui d'un diagramme de classes qui contient toutes les informations telles que les classes, les méthodes, les associations et les propriétés.

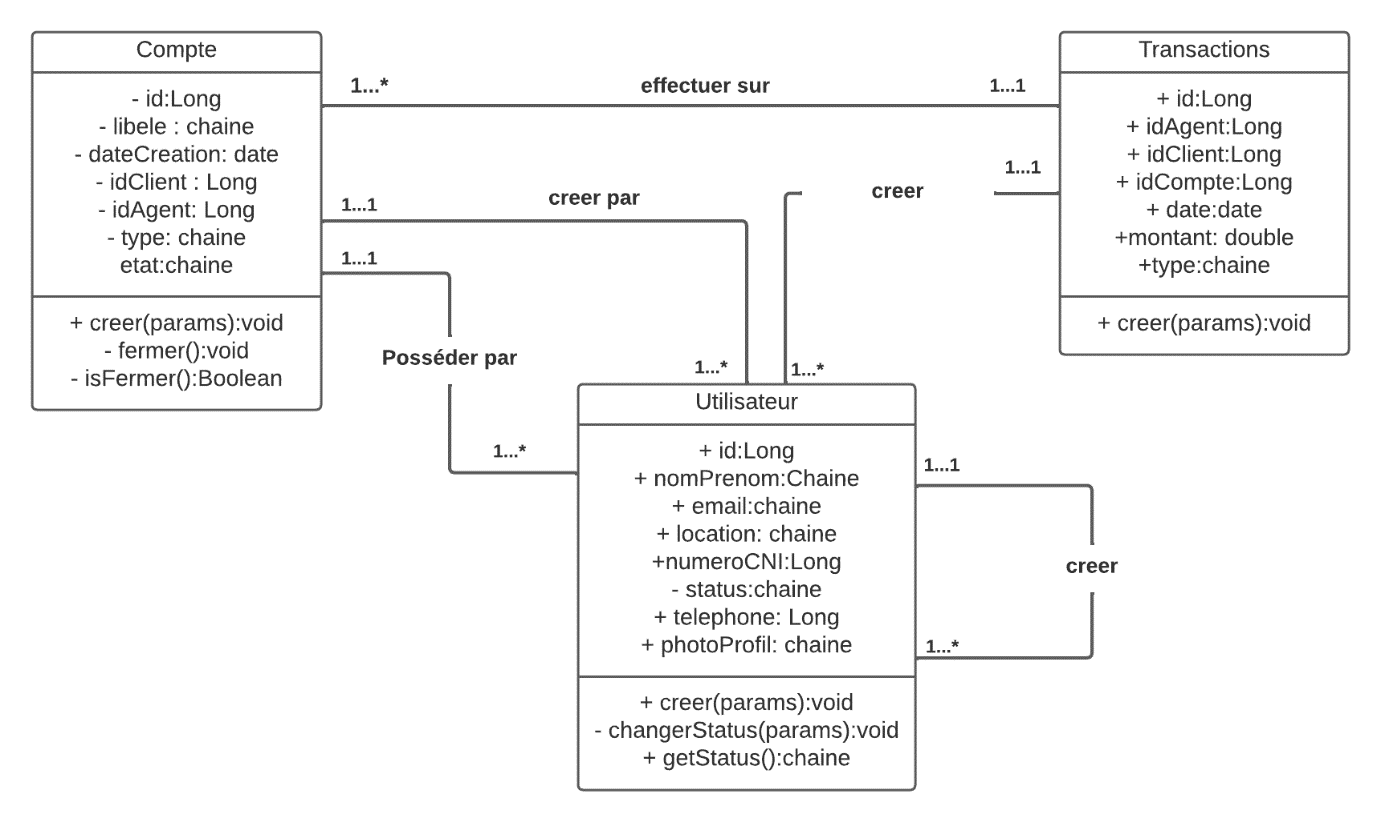


Figure 08 : diagramme de Classe

# CHAPITRE III : DOSSIER DE RÉALISATION

Dans le dossier de réalisation, nous présenterons les différents outils nécessaires à la mise en œuvre de la solution. Pour la réalisation du service web et de la plateforme, nous allons utiliser les technologies récentes tout en gardant à l’esprit que notre application doit être facilement manipulable (User friendly).

Après avoir élaboré la conception de notre application, nous abordons dans ce chapitre le dernier volet de ce rapport, qui a pour objectif d'exposer la phase de réalisation. La phase de réalisation est considérée comme étant la concrétisation finale de toute la méthode de conception.

Nous menons tout d’abord une étude technique où nous décrivons les ressources logicielles utilisées dans le développement de notre projet. Nous présentons en premier lieu notre choix de l’environnement de travail, où nous spécifions l’environnement matériel et logiciel qu‘on a utilisé pour réaliser notre application puis nous détaillons l’architecture, aussi nous présentons quelques interfaces réalisées pour illustrer le fonctionnement de quelques activités du système.

## L’ÉTUDE TECHNIQUE

L'étude technique est une phase d'adaptation de conception à l'architecture technique. Elle a pour objectif de décrire au plan fonctionnel la solution à réaliser d'une manière détaillée ainsi que la description des traitements. Cette étude, qui suit l'étude détaillée, constitue le complément de spécification informatique nécessaire pour assurer la réalisation du futur système. Cette étude permet également de déterminer :

* La structure informatique de la base de données
* L'architecture des programmes
* La structure de chaque programme et l'accès aux données.

### I.1) Environnement de réalisation

Pour la réalisation de notre application, nous avons eu recours à plusieurs moyens matériels et logiciels :

#### I.1.1) Matériels de base :

Tableau 17 : Matériels de base

|  |  |  |
| --- | --- | --- |
| **Matériel** | **Caractéristique** | **Utilité** |
| Ordinateur portable | i5 5e Gene 8Go Ram | Machine de développement et de teste de la version web |
| 3 téléphone mobile Android | * 5 ‘’ pouces * 7’’ pouces * Tablette | Les téléphone de taille d’écran différente sont là pour valider la compatibilité des vue su système sur diffèrent écran. |

#### I.1.2) Choix des langages de développement et de SGBD :

* + - 1. Application mobile
* Frontend
* XML :



Figure 09 : logo XML

L'Extensible Markup Language, généralement appelé XML, « langage de balisage extensible » en français, est un [métalangage](https://fr.wikipedia.org/wiki/M%C3%A9talangage) informatique de [balisage](https://fr.wikipedia.org/wiki/Langage_de_balisage) générique qui est un sous-ensemble du [Standard Generalized Markup Language](https://fr.wikipedia.org/wiki/Standard_Generalized_Markup_Language) (SGML). Sa syntaxe est dite « extensible » car elle permet de définir différents langages avec pour chacun son vocabulaire et sa grammaire, comme [XHTML](https://fr.wikipedia.org/wiki/XHTML), [XSLT](https://fr.wikipedia.org/wiki/XSLT), [RSS](https://fr.wikipedia.org/wiki/RSS), [SVG](https://fr.wikipedia.org/wiki/SVG)… Elle est reconnaissable par son usage des [chevrons](https://fr.wikipedia.org/wiki/Chevron_(typographie)) (<, >) encadrant les noms des balises. L'objectif initial de XML est de faciliter l'échange automatisé de contenus complexes ([arbres](https://fr.wikipedia.org/wiki/Arbre_enracin%C3%A9), texte enrichi, etc.) entre [systèmes d'informations](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27information) hétérogènes ([interopérabilité](https://fr.wikipedia.org/wiki/Interop%C3%A9rabilit%C3%A9_en_informatique)).

https://fr.wikipedia.org/wiki/Extensible\_Markup\_Language

* Backend
* JAVA



Figure 10 : logo Java

Java permet de développer des applications [client-serveur](https://fr.wikipedia.org/wiki/Client-serveur). Côté client, les [applets](https://fr.wikipedia.org/wiki/Appliquette) sont à l’origine de la notoriété du langage. C’est surtout côté serveur que Java s’est imposé dans le milieu de l’entreprise grâce aux [servlets](https://fr.wikipedia.org/wiki/Servlet), le pendant serveur des applets, et plus récemment les JSP ([JavaServer Pages](https://fr.wikipedia.org/wiki/JavaServer_Pages)) qui peuvent se substituer à [PHP](https://fr.wikipedia.org/wiki/PHP), [ASP](https://fr.wikipedia.org/wiki/Active_Server_Pages) et [ASP.NET](https://fr.wikipedia.org/wiki/ASP.NET).

https://fr.wikipedia.org/wiki/Java\_(langage)

* Firebase

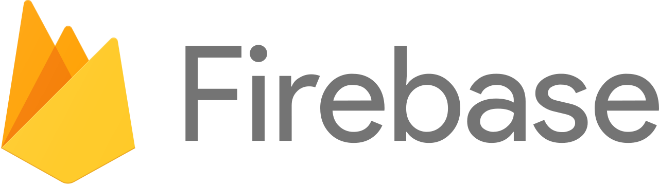


Figure 11 : logo Firebase

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application ([Android](https://fr.wikipedia.org/wiki/Android), [iOS](https://fr.wikipedia.org/wiki/IOS_(Apple)), [Javascript](https://fr.wikipedia.org/wiki/Javascript), [Node.js](https://fr.wikipedia.org/wiki/Node.js), [Java](https://fr.wikipedia.org/wiki/Java_(technique)), [Unity](https://fr.wikipedia.org/wiki/Unity_(moteur_de_jeu)), [PHP](https://fr.wikipedia.org/wiki/PHP), [C++](https://fr.wikipedia.org/wiki/C%2B%2B) ...). Il propose d'héberger en [NoSQL](https://fr.wikipedia.org/wiki/NoSQL) et en temps réel des bases de données, du contenu, de l'[authentification sociale](https://fr.wikipedia.org/wiki/Social_login) ([Google](https://fr.wikipedia.org/wiki/Google), [Facebook](https://fr.wikipedia.org/wiki/Facebook), [Twitter](https://fr.wikipedia.org/wiki/Twitter) et [Github](https://fr.wikipedia.org/wiki/Github)), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. Lancé en 2011 sous le nom d'Envolve, par Andrew Lee et par James Tempslin, le service est racheté par [Google](https://fr.wikipedia.org/wiki/Google) en octobre 2014. Il appartient aujourd'hui à la maison mère de [Google](https://fr.wikipedia.org/wiki/Google) : [Alphabet](https://fr.wikipedia.org/wiki/Alphabet_(entreprise)).

https://fr.wikipedia.org/wiki/Firebase

#### I.1.3) Application web

* Frontend
* ReactJs

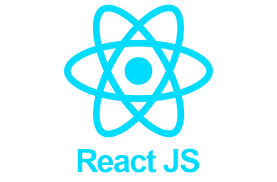


Figure 12 : logo React JS

React (également connu sous le nom de React.js ou ReactJS ) est une [bibliothèque JavaScript](https://en.wikipedia.org/wiki/JavaScript_library)[frontale](https://en.wikipedia.org/wiki/Front_end_and_back_end)[gratuite et open source](https://en.wikipedia.org/wiki/Free_and_open-source_software)[[3]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-react-3) pour la création [d'interfaces utilisateur](https://en.wikipedia.org/wiki/User_interfaces) basées sur des composants d'interface utilisateur. Il est maintenu par [Meta](https://en.wikipedia.org/wiki/Meta_Platforms) (anciennement Facebook) et une communauté de développeurs individuels et d'entreprises. [[4]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-4)[[5]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-5)[[6]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-6) React peut être utilisé comme base dans le développement d' applications [monopage](https://en.wikipedia.org/wiki/Single-page_application) , mobiles ou rendues par un serveur avec des frameworks comme [Next.js.](https://en.wikipedia.org/wiki/Next.js)Cependant, React ne concerne que la gestion de l'état et le rendu de cet état au [DOM](https://en.wikipedia.org/wiki/Document_Object_Model) , donc la création d'applications React nécessite généralement l'utilisation de bibliothèques supplémentaires pour le routage, ainsi que certaines fonctionnalités côté client.

* **JavaScript**

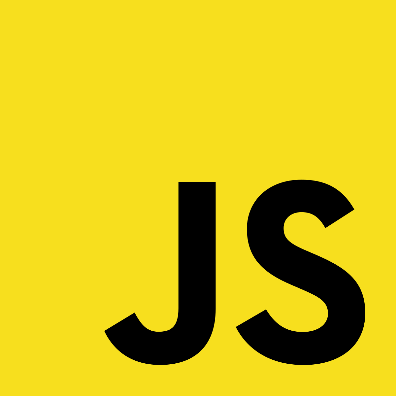


Figure 13 : logo JavaScript

JavaScript est un [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) de [scripts](https://fr.wikipedia.org/wiki/Langage_de_script) principalement employé dans les [pages web](https://fr.wikipedia.org/wiki/Pages_web) interactives et à ce titre est une partie essentielle des [applications web](https://fr.wikipedia.org/wiki/Application_web). Avec les langages [HTML](https://fr.wikipedia.org/wiki/HTML) et [CSS](https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade), JavaScript est au cœur des langages utilisés par les [développeurs web](https://fr.wikipedia.org/wiki/D%C3%A9veloppeur_web)[3](https://fr.wikipedia.org/wiki/JavaScript#cite_note-3). Une grande majorité des [sites web](https://fr.wikipedia.org/wiki/Site_web) l'utilisent[4](https://fr.wikipedia.org/wiki/JavaScript#cite_note-deployedstats-4), et la majorité des [navigateurs web](https://fr.wikipedia.org/wiki/Navigateur_web) disposent d'un [moteur JavaScript](https://fr.wikipedia.org/wiki/Moteur_JavaScript)[5](https://fr.wikipedia.org/wiki/JavaScript#cite_note-5) pour l'[interpréter](https://fr.wikipedia.org/wiki/Interpr%C3%A8te_(informatique)). JavaScript est un [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) de [scripts](https://fr.wikipedia.org/wiki/Langage_de_script) principalement employé dans les [pages web](https://fr.wikipedia.org/wiki/Pages_web) interactives et à ce titre est une partie essentielle des [applications web](https://fr.wikipedia.org/wiki/Application_web). Avec les langages [HTML](https://fr.wikipedia.org/wiki/HTML) et [CSS](https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade), JavaScript est au cœur des langages utilisés par les [développeurs web](https://fr.wikipedia.org/wiki/D%C3%A9veloppeur_web)[3](https://fr.wikipedia.org/wiki/JavaScript#cite_note-3). Une grande majorité des [sites web](https://fr.wikipedia.org/wiki/Site_web) l'utilisent[4](https://fr.wikipedia.org/wiki/JavaScript#cite_note-deployedstats-4), et la majorité des [navigateurs web](https://fr.wikipedia.org/wiki/Navigateur_web) disposent d'un [moteur JavaScript](https://fr.wikipedia.org/wiki/Moteur_JavaScript)[5](https://fr.wikipedia.org/wiki/JavaScript#cite_note-5) pour l'[interpréter](https://fr.wikipedia.org/wiki/Interpr%C3%A8te_(informatique)).

* **HTML**



Figure 14 : logo HTML

HTML est le langage de balisage standard pour les documents conçus pour être affichés dans un navigateur Web. Il peut être assisté par des technologies telles que la feuille de style en cascade (CSS) et des langages de script tels que JavaScript.

* **Sass**



Figure 15 : logo Sass

Sass (Syntactically awesome stylesheets) est un [langage de script](https://fr.wikipedia.org/wiki/Langage_de_script) [préprocesseur](https://fr.wikipedia.org/wiki/Pr%C3%A9processeur) qui est [compilé](https://fr.wikipedia.org/w/index.php?title=Langage_compil%C3%A9&action=edit&redlink=1) ou [interprété](https://fr.wikipedia.org/wiki/Interpr%C3%A8te_(informatique)) en [CSS](https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade) (Cascading Style Sheets). SassScript est le langage de script en lui-même.

Sass est disponible en deux [syntaxes](https://fr.wikipedia.org/w/index.php?title=Syntaxe_(Informatique)&action=edit&redlink=1). La syntaxe originale, appelée "syntaxe indentée"[3](https://fr.wikipedia.org/wiki/Sass_(langage)#cite_note-3) qui utilise l'[indentation](https://fr.wikipedia.org/wiki/Style_d%27indentation) pour séparer les blocs de code et les [sauts de ligne](https://fr.wikipedia.org/wiki/Fin_de_ligne) pour les séparer les directives. La nouvelle syntaxe, "SCSS", utilise les mêmes séparateurs de blocs que CSS. Les fichiers de la syntaxe indentée et SCSS utilisent respectivement les extensions. sass et .scss.

* **GitHub**



Figure 16 : logo GitHub

GitHub (/[ɡ](https://fr.wikipedia.org/wiki/API_%C9%A1)[ɪ](https://fr.wikipedia.org/wiki/API_%C9%AA)[t](https://fr.wikipedia.org/wiki/API_t)[h](https://fr.wikipedia.org/wiki/API_h)[ʌ](https://fr.wikipedia.org/wiki/API_%CA%8C)[b](https://fr.wikipedia.org/wiki/API_b)/, entreprise GitHub, Inc.) est un service web d'[hébergement](https://fr.wikipedia.org/wiki/H%C3%A9bergeur_web) et de gestion de [développement de logiciels](https://fr.wikipedia.org/wiki/D%C3%A9veloppement_de_logiciel), utilisant le [logiciel de gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions) [Git](https://fr.wikipedia.org/wiki/Git). Ce site est développé en [Ruby on Rails](https://fr.wikipedia.org/wiki/Ruby_on_Rails) et [Erlang](https://fr.wikipedia.org/wiki/Erlang_(langage)) par Chris Wanstrath, PJ Hyett et Tom Preston-Werner. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de [logiciels libres](https://fr.wikipedia.org/wiki/Logiciels_libres).

Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet. Le site est devenu le plus important dépôt de code au monde, utilisé comme dépôt public de projets libres ou dépôt privé d'entreprises.

#### I.1.4) Outil de développement

* Visual Studio Code : Editeur de texte
* Android Studio : Environnement de développement intégré (IDE) utiliser pour le développement des applications Android.
* Nodejs



Figure 17 : logo node js

Node.js est une [plateforme logicielle](https://fr.wikipedia.org/wiki/Plate-forme_(informatique)) [libre](https://fr.wikipedia.org/wiki/Logiciel_libre) en [JavaScript](https://fr.wikipedia.org/wiki/JavaScript), orientée vers les applications [réseau](https://fr.wikipedia.org/wiki/R%C3%A9seau_informatique) [évènementielles](https://fr.wikipedia.org/wiki/Programmation_%C3%A9v%C3%A9nementielle) hautement [concurrentes](https://fr.wikipedia.org/wiki/Programmation_concurrente) qui doivent pouvoir [monter en charge](https://fr.wikipedia.org/wiki/Scalability).

Elle utilise la [machine virtuelle](https://fr.wikipedia.org/wiki/Machine_virtuelle) [V8](https://fr.wikipedia.org/wiki/V8_(moteur_JavaScript)), la librairie [libuv](https://en.wikipedia.org/wiki/libuv) pour sa [boucle d'évènements](https://en.wikipedia.org/wiki/Event_loop), et implémente sous [licence MIT](https://fr.wikipedia.org/wiki/Licence_MIT) les spécifications [CommonJS](https://fr.wikipedia.org/wiki/CommonJS).

Parmi les modules natifs de Node.js, on retrouve [http](https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol) qui permet le développement de [serveur HTTP](https://fr.wikipedia.org/wiki/Serveur_HTTP). Ce qui autorise, lors du déploiement de sites internet et d'applications web développés avec Node.js, de ne pas installer et utiliser des [serveurs webs](https://fr.wikipedia.org/wiki/Serveur_informatique) tels que [Nginx](https://fr.wikipedia.org/wiki/Nginx) ou [Apache](https://fr.wikipedia.org/wiki/Apache_HTTP_Server).

Concrètement, Node.js est un environnement bas niveau permettant l'exécution de [JavaScript](https://fr.wikipedia.org/wiki/JavaScript) côté serveur.

* **Adobe Photoshop CS6**



Figure 18 : logo PhotoShop

Photoshop est un [logiciel](https://fr.wikipedia.org/wiki/Logiciel) de [retouche](https://fr.wikipedia.org/wiki/Retouche_d%27image), de [traitement](https://fr.wikipedia.org/wiki/Traitement_d%27images) et de [dessin assisté par ordinateur](https://fr.wikipedia.org/wiki/Dessin_assist%C3%A9_par_ordinateur), lancé en 1990 puis en 1992 pour les [systèmes d'exploitations](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) MacOS et Windows.

Édité par la société [Adobe](https://fr.wikipedia.org/wiki/Adobe_Systems), il est principalement utilisé pour le traitement des [photographies numériques](https://fr.wikipedia.org/wiki/Photographie_num%C3%A9rique) et sert également à la création [ex nihilo](https://fr.wikipedia.org/wiki/Ex_nihilo) d’images.

Il travaille essentiellement sur [images matricielles](https://fr.wikipedia.org/wiki/Image_matricielle) car les images sont constituées d’une grille de points appelés [pixels](https://fr.wikipedia.org/wiki/Pixel). L’intérêt de ces images est de reproduire des gradations subtiles de couleurs. https://fr.wikipedia.org/wiki/Adobe\_Photoshop

* **Microsoft office Word 2016**



Figure 19 : logo Microsoft Word

Microsoft Word, qui est un logiciel de traitement de texte couvre deux notions, assez différentes en pratique : Un éditeur de textes interactif et un compilateur pour un langage de mise en forme de textes.

* **Microsoft office Power point 2016**



Figure 20 : logo Microsoft PowerPoint

Microsoft Office PowerPoint est un créateur de présentations (succession de diapositives). Il est utilisé pour créer des présentations avec du texte, avec des images, sons, vidéos et autres objets

### I.2) Modélisation physique des données

Pour aboutir à une description technique de la base de données dans le système utilisé, il faut transformer le MLC qui est déduit du diagramme de classes, puis optimisé en fonction des traitements qui vont utiliser le modèle.

La transformation au niveau physique doit tenir compte des caractéristiques et des contraintes du système utilisé afin d’obtenir un modèle physique réalisable et performant. Dans cette partie, nous décrivons la modélisation physique des données.

Notre base donne étant du NO-SQL qui comporte plusieurs modèles de représentation des données.

Schéma de l’architecture du modèle de représentation « collection - document »

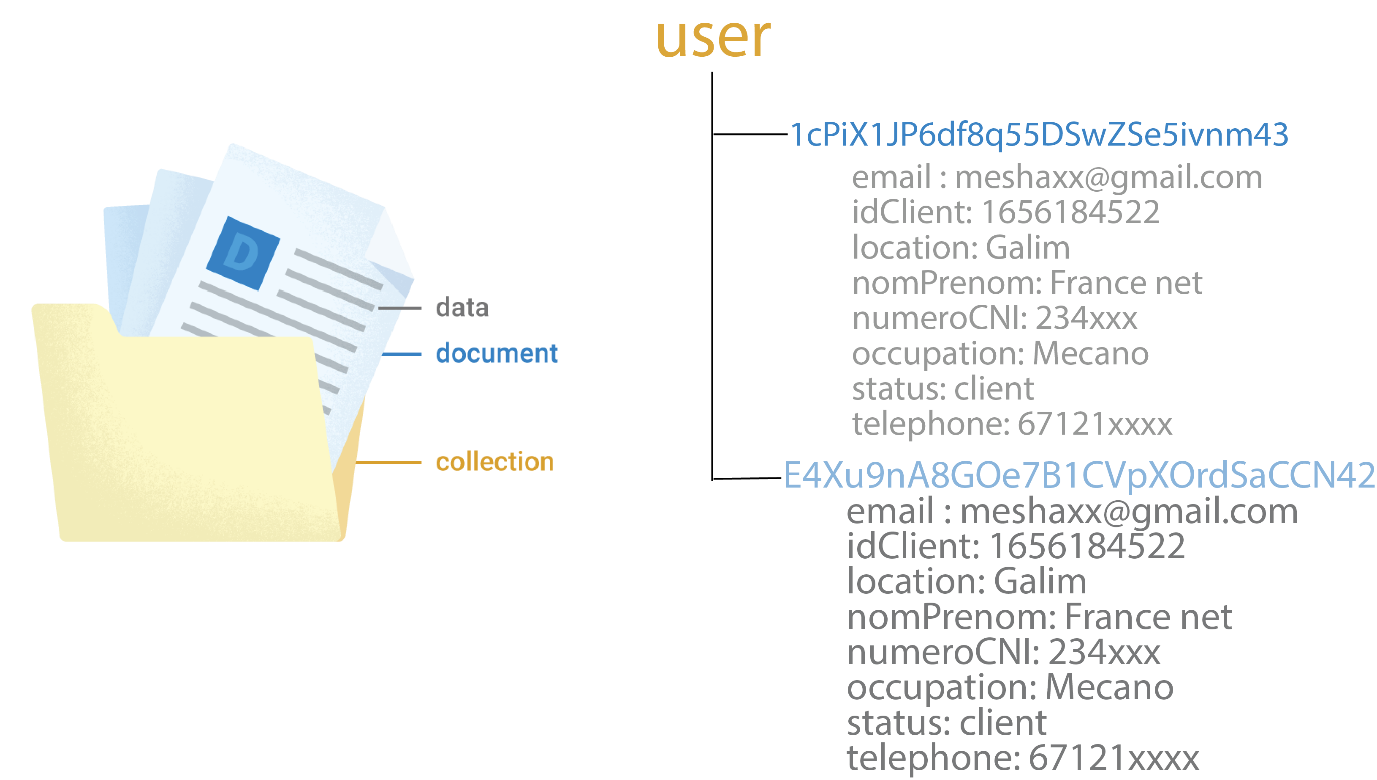


Figure 21 : représentation de la collection des utilisateur

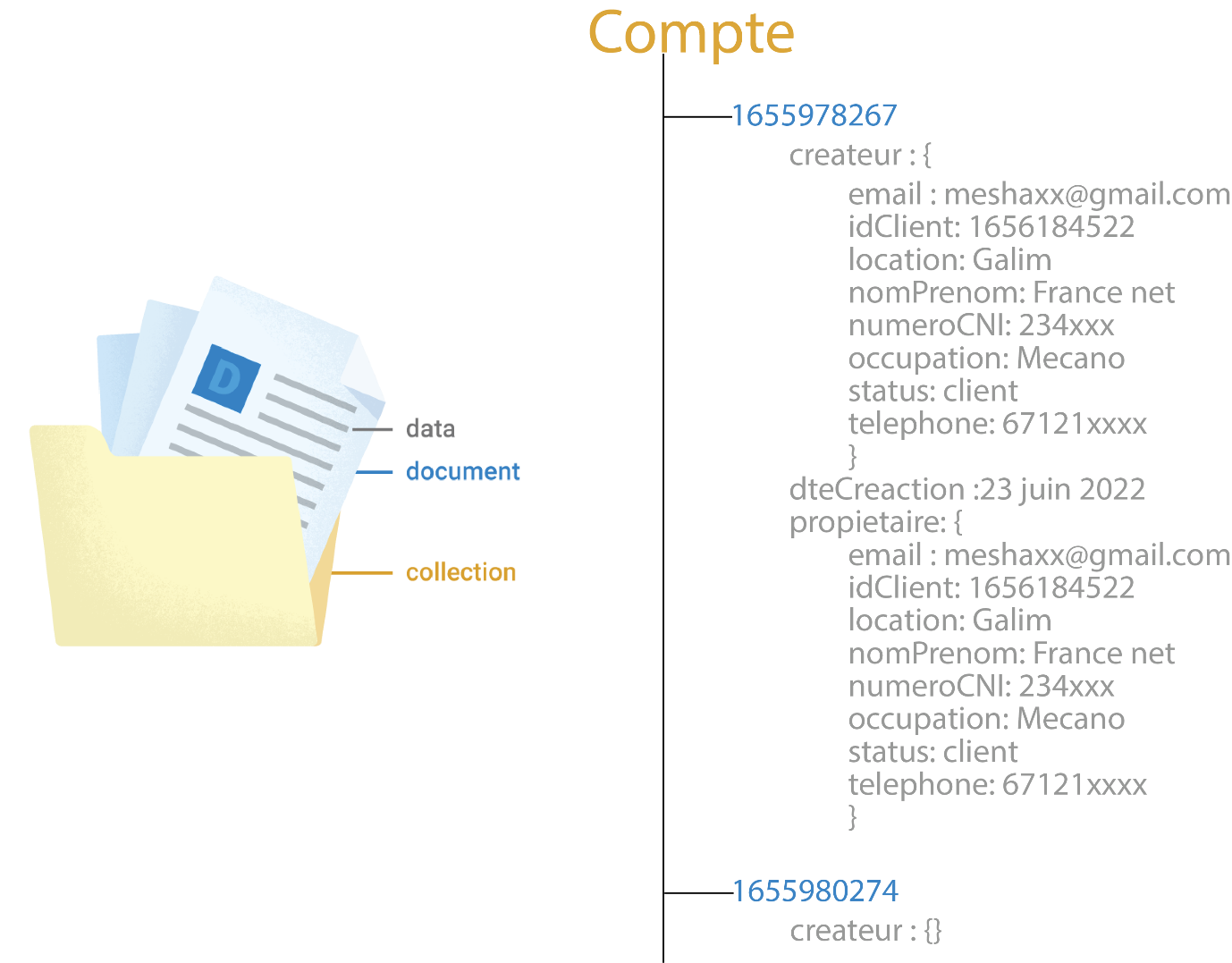


Figure 22 : représentation de la collection des Comptes

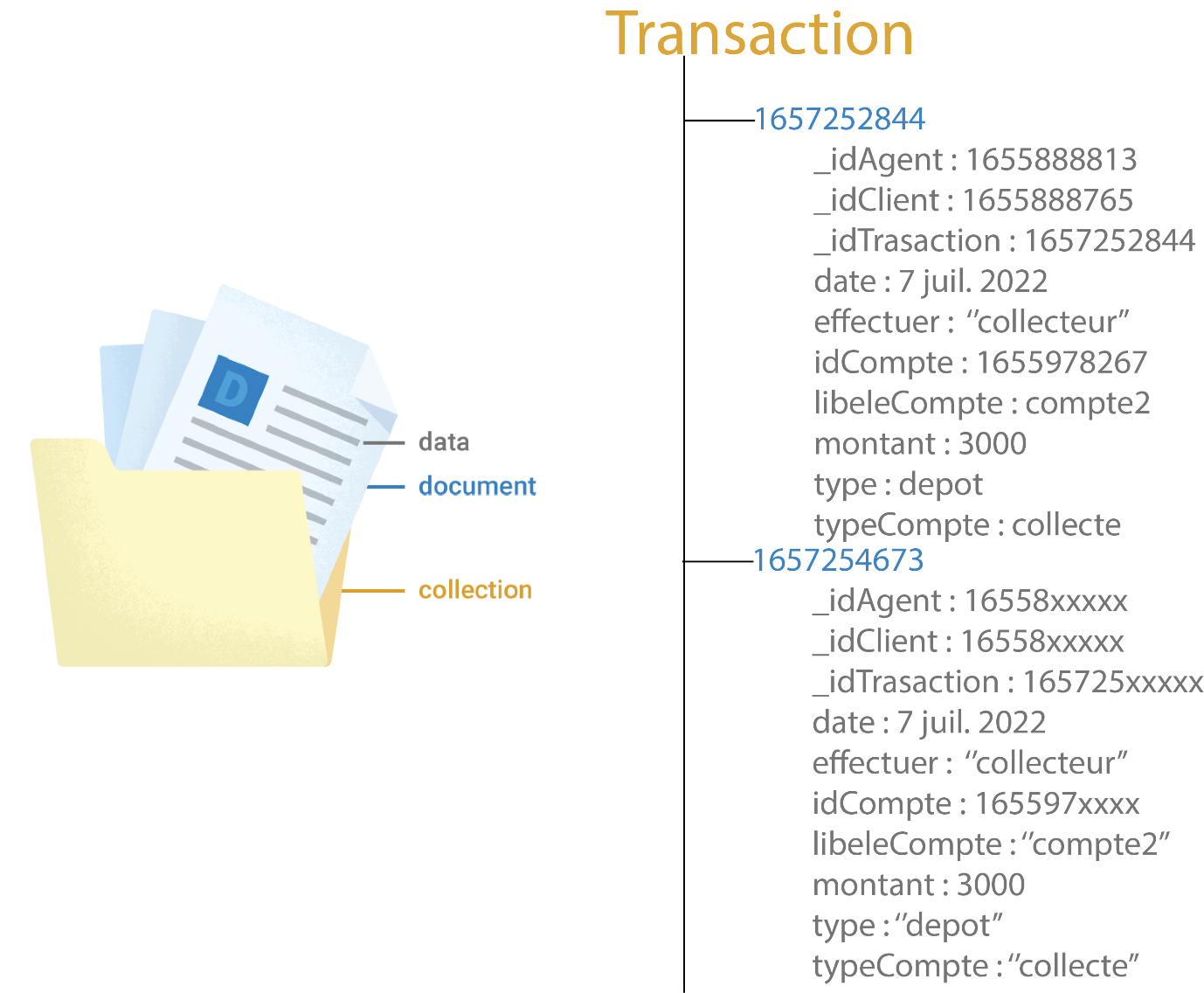


Figure 23 : représentation de la collection des Transaction

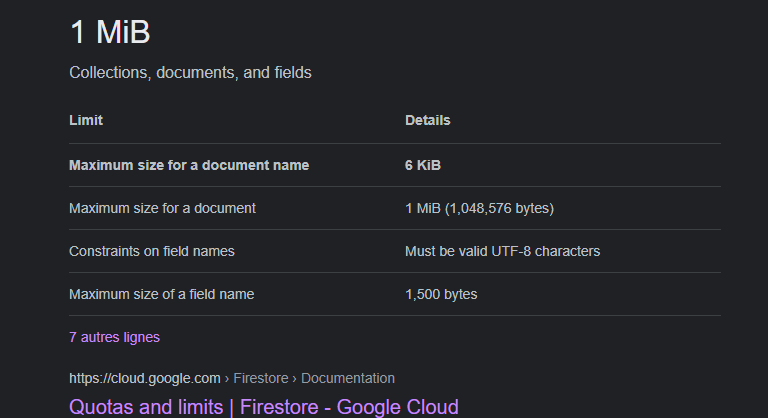


Figure 24 : règle de création d’un document

Ainsi pour notre nous avons trois collections : user (collection des utilisateurs), Compte (collection des comptes), Transactions (collections des transactions).

Tableau 17 : Modelé physique de donnés.

|  |  |
| --- | --- |
| **Collection** | **Champs** |
| user | email |
| idClient |
| location |
| nomPrenom |
| numeroCNI |
| Occupation |
| status |
| telephone |
| Compte | createur : objet de la classe user |
| propietaire : objet de la classe user |
| type |
| dteCreaction |
| libele |
| idCompte |
|  |
| Transaction | \_idAgent |
| \_idClient |
| \_idTrasaction |
| date |
| effectuer |
| idCompte |
| libeleCompte |
| montant |
|  | Type |
| typeCompte |

## PRODUCTION DES PROGRAMMES

### II.1) Descriptif du produit

Notre projet est constitué de 3 couches pour la version mobile et d’une couche web.

#### II.1.1) Version mobile

* Couche des clients
* Couche des collecteurs
* Couche de l’administration

#### II.1.2) Version web

* Couche administrateur

### II.2) Architecture

L’architecture de notre application est de type client-serveur, où un ordinateur/mobile interagit avec un serveur distant.

Dans la réalisation de notre projet, nous avons opté pour une architecture MVC afin de garantir une assurance de la maintenabilité, la modularité de l’application et la rapidité de développement.

MVC littéralement Modèle Vue Contrôleur est une architecture qui organise l'interface Homme-Machine d'une manière à ce que le développement puisse se faire en couches indépendantes. Il impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle de données, le contrôleur et la vue

### II.3) Charte graphique

#### II.3.1) Arrière-plan et texte

Nous avons opté pour un fond blanc. En effet, les études montrent que les sites conçus avec un fond blanc connaissent plus de succès que les autres. La plupart des moteurs de recherche ou des boutiques virtuelles s’affichent sur fond blanc. Le fond blanc renvoie au papier. Il conforte le lecteur et communique une impression de sérieux, de sobriété, de détachement et d’objectivité.

Nous avons écrit le texte en noir ou en couleurs sombres, pour plus de confort pour le lecteur.

#### II.3.2) Usage des couleurs

Étant une application représentative notre système doit utiliser les couleurs respectant déjà la charte graphique de la microfinance.

Donc pour les objets primaires nous avons choisi une couleur : orange

Et le noir comme couleur secondaire : le noir

Le blanc pour mixer les contenue pour apporter un flow plus dégager et donner aux utilisateurs un aspect convivial.

#### II.3.3) Charte graphique stable et robuste

Notre charte graphique est basée sur la technique des feuilles de style CSS ce qui la rend facile à manipuler :

* Il vous suffira de changer les définitions de style figurant dans les feuilles de style CSS pour que la présentation de notre site soit mise à jour.
* Le nombre de page web en ligne n’est pas statique (on peut changer le contenu textuel et visuel du site et notre charte graphique reste stable et robuste).
* Le mode nuit pour réduire la lumière forte et protéger les yeux des utilisateurs.

## PHASE DE DÉPLOIEMENT

### III.1) Diagramme de déploiement

Le diagramme de déploiement peut être mis en œuvre dès la phase de conception pour documenter l’architecture physique du système.

#### III.1.1) Formalisme

Le diagramme de déploiement peut être mis en œuvre dès la phase de conception pour documenter l’architecture physique du système.

|  |  |  |
| --- | --- | --- |
| **Élément** | **Description** | **Représentation** |
| Les composants | Les objets sont les instances de classe et sont rangés horizontalement. |  |
| Les nœuds | Un nœud représente l’ensemble d’équipements matériels du système. Cette entité est représentée par un cube tridimensionnel. |  |
| Les dépendances | Une dépendance est utilisée pour mobiliser la relation entre deux composants. La notion utilisée pour cette relation de dépendance est une flèche de pointilles. |  |
| Association | Relier les nœuds |  |

#### III.1.2) Diagramme de déploiement

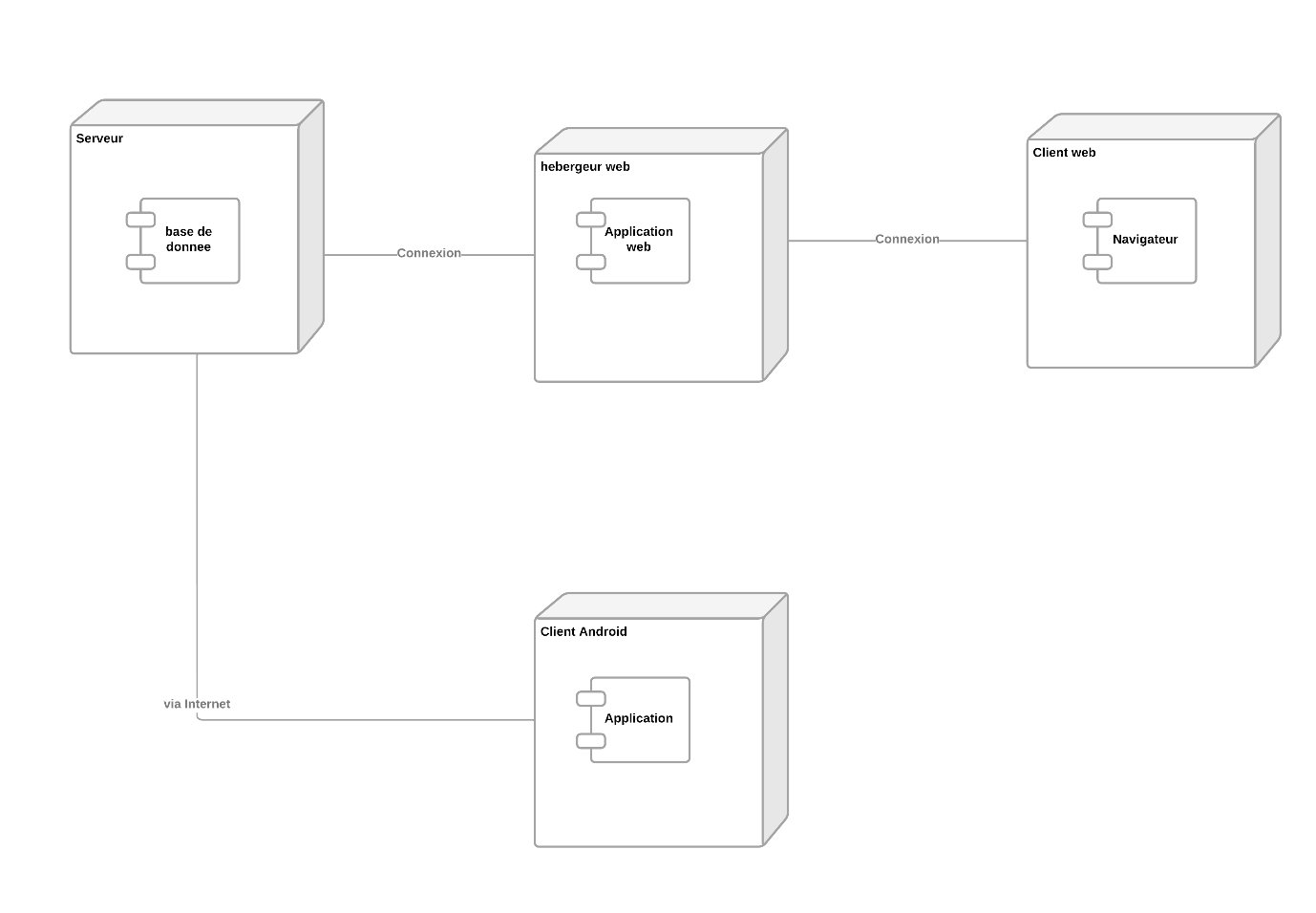


Figure 25 : Diagramme de déploiement

Conclusion

# PARTIE III – BILAN

# Introduction

Cette partie est dédié au bilan des différentes taches effectuer au courant du stage, où nous avons présenté l’application résultant de notre Project et les différents apports.

# CHAPITRE 1 : PRÉSENTATION DE L’APPLICATION

Après les phases d’étude de l’existant, la conception et la modélisation nous avons développé les interfaces de notre application.

Nous allons présenter ici la partie Back-office et FrontOffice de l’application ainsi que ses différentes fonctionnalités y afférentes.

## LES VUES

### I.1) Version mobile

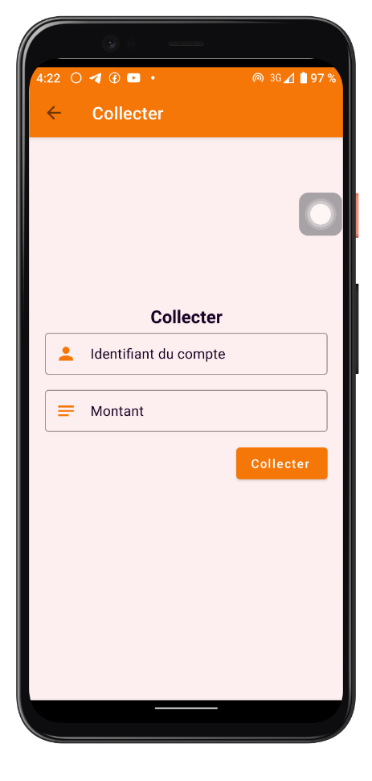
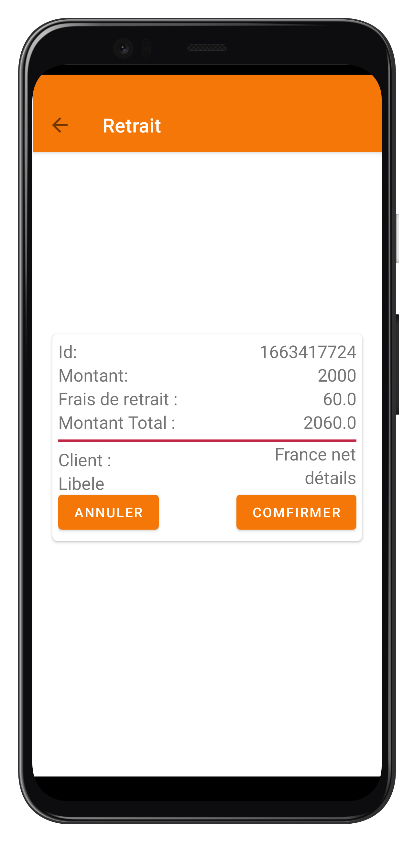
 

Figure 26 : vues : formulaires de (collecte, ouverture de compte, confirmer un retrait)

Ces vues présentes comment les utilisateurs doivent respecter les entrée les donnes dans le système.

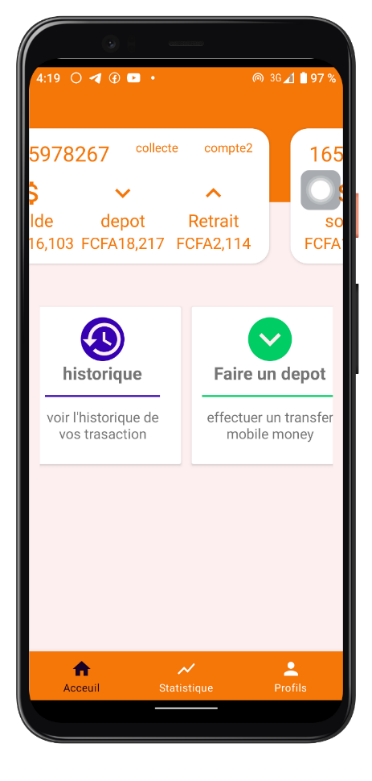


Figure 27 : vues : Les différent tableaux de bord : collecteur, client, administrateur

Le tableau de bords est différent selon le type d’utisateur car les utilisateur non pas les même niveaux d’acces au donnees.

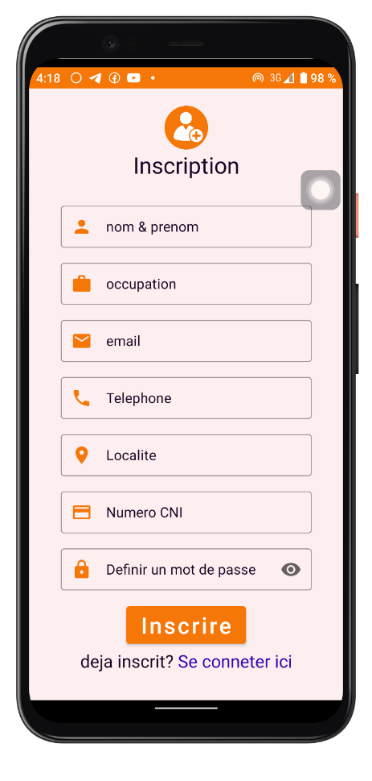


Figure 28 : vues : formulaires d’inscription et de connexion

Pour l’écran d’inscription, n’importe qui peut s’inscrire. Il suffit d’entrer les information valide. Pour l’écran de connexion par contre il faut déjà être inscrit. Même ayant entrés les informations valide il est possible si l’utilisateur est banni de ne pas avoir accès au tableau de bords.

### I.2) Version web

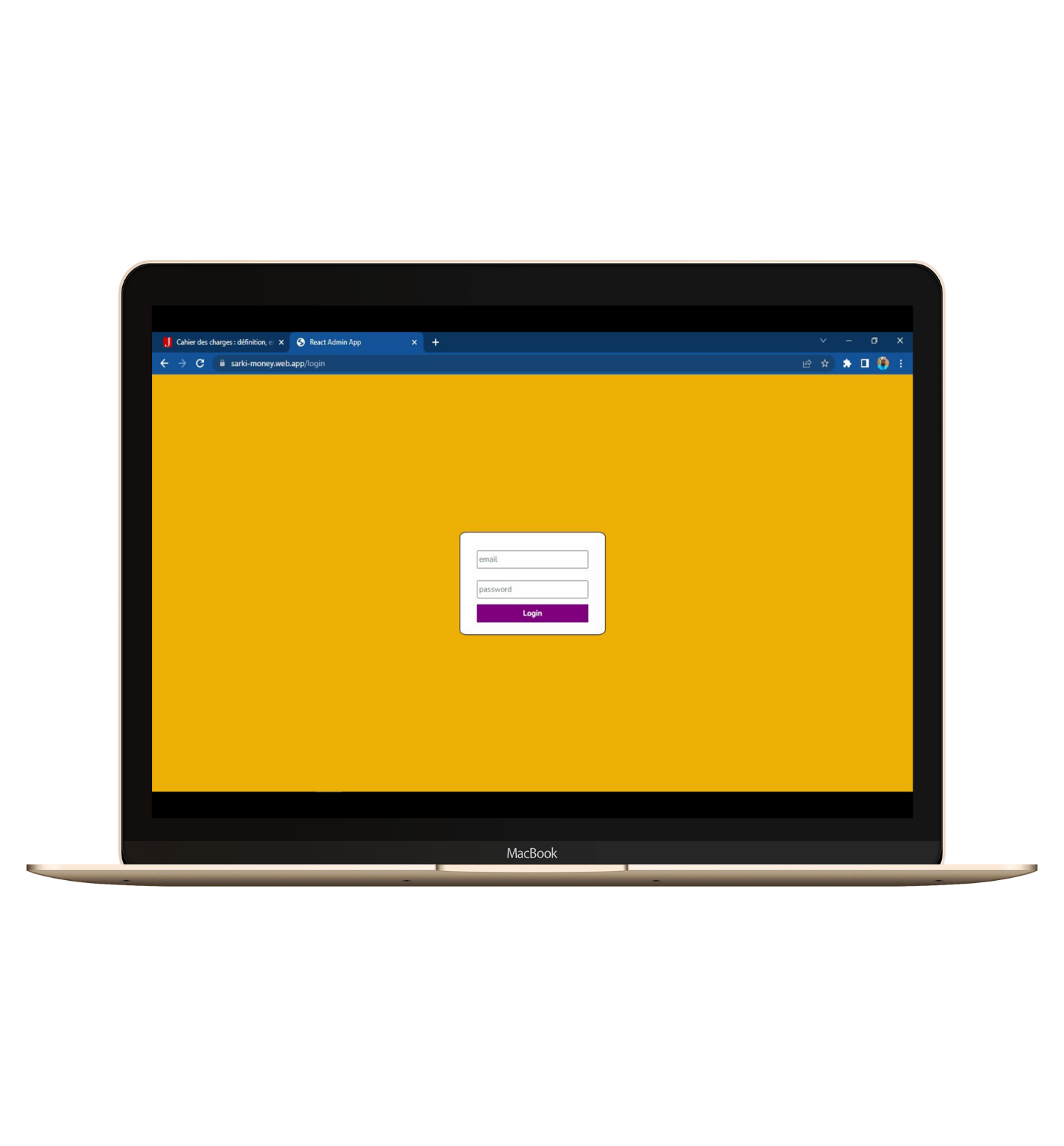


Figure 29 : Vue : formulaires de Connexion réserver à l’administrateur

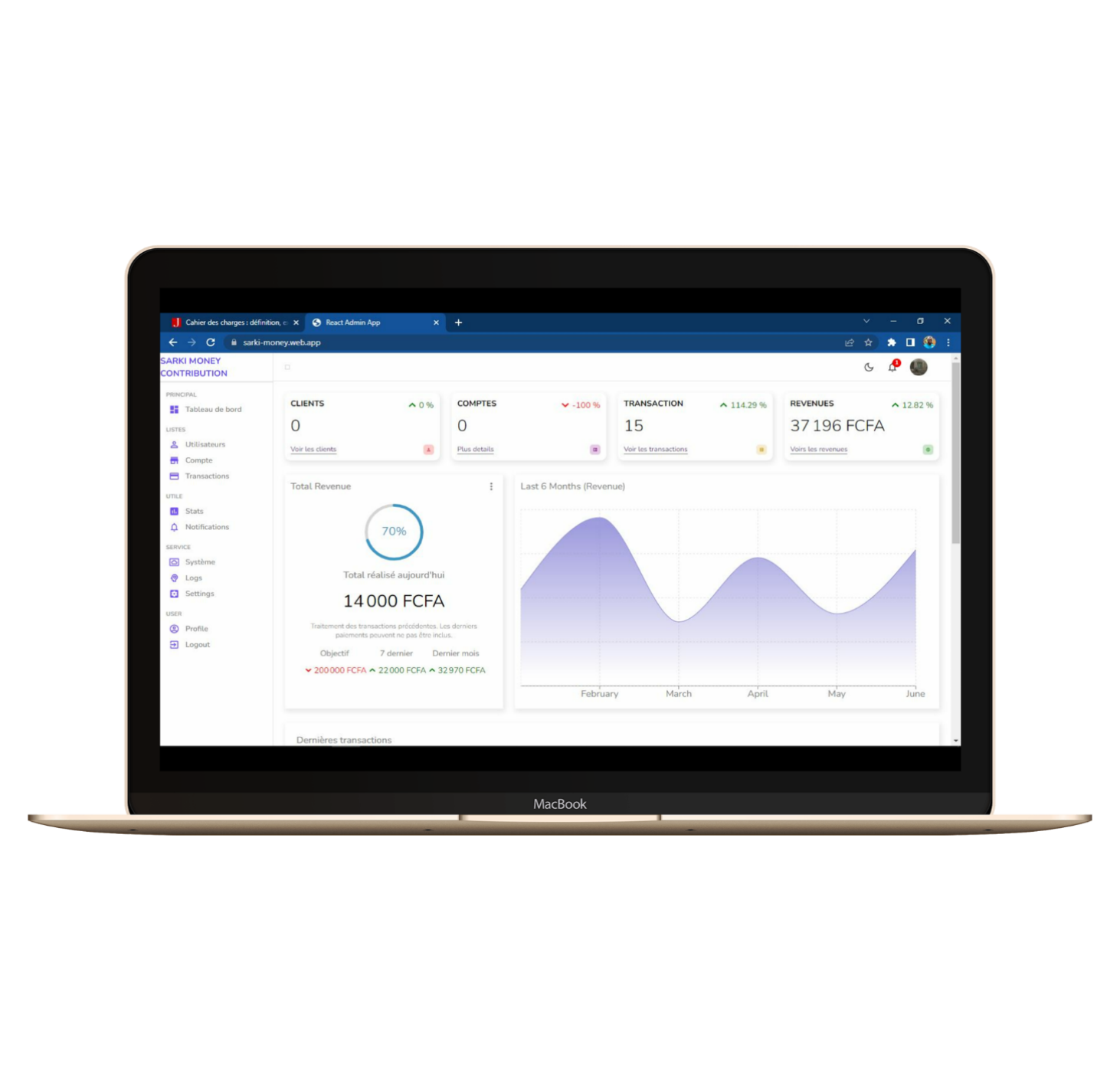


Figure 30 : vues : tableau de bord de l’administrateur sur web

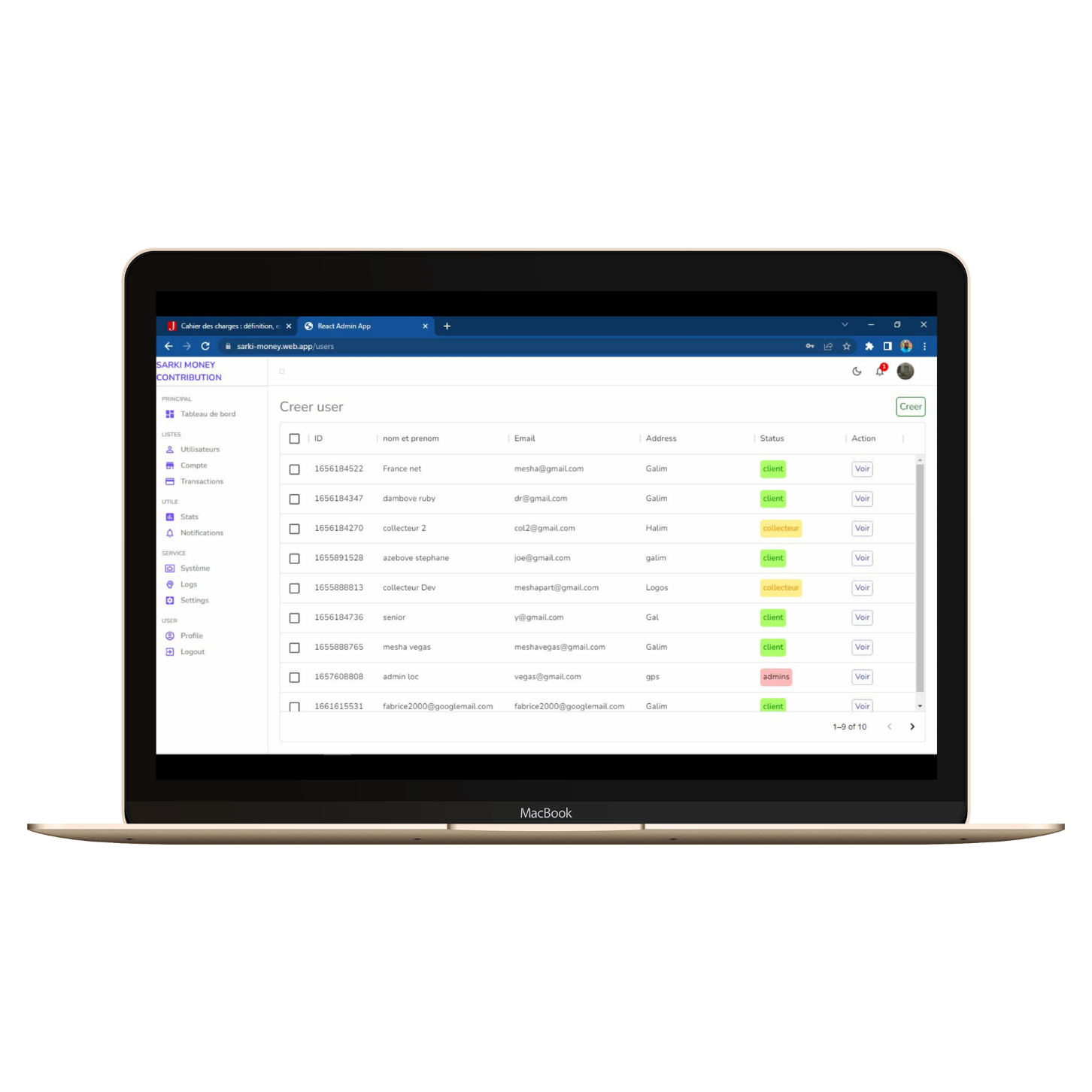


Figure 31 : vues liste des utilisateurs

#### I.2.1) Les modèles

Pour le modèle de donne est unique puisque. Les utilisateurs mobiles et web ont une seule base de données.

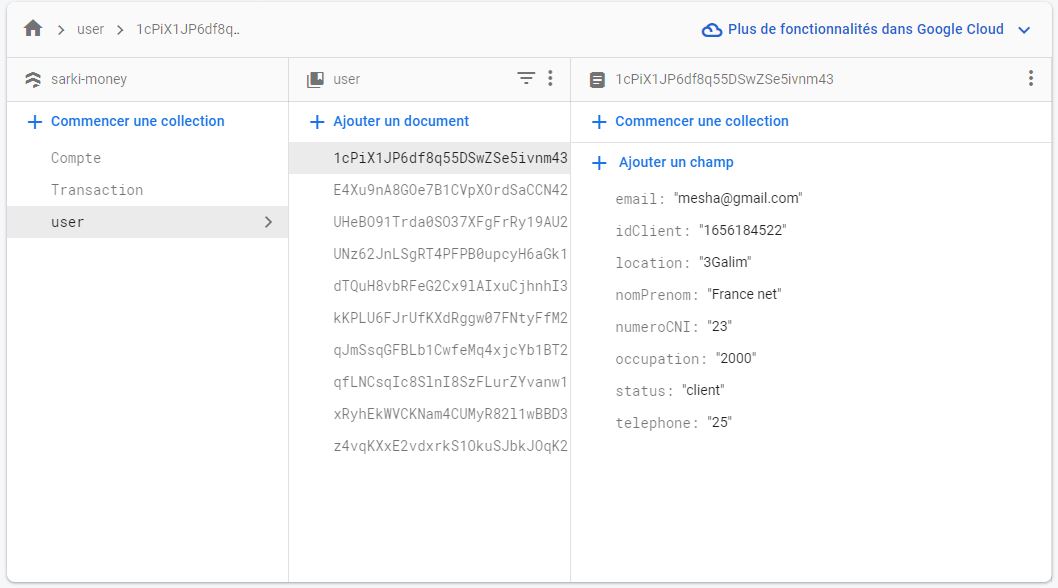


Figure 32 : collection des utilisateurs

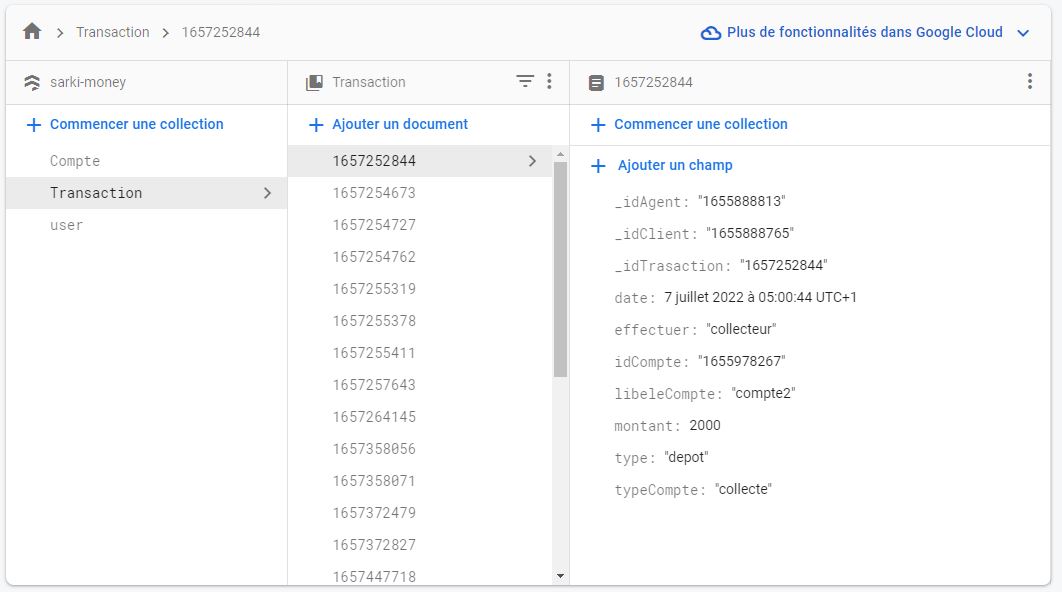


Figure 33 : collection des transactions

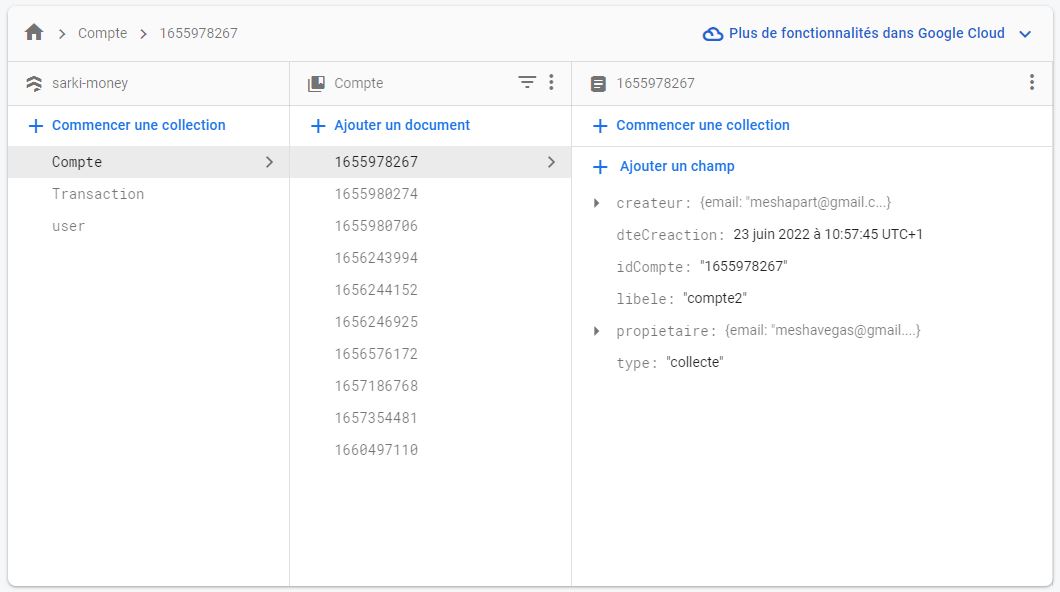


Figure 32 : collection des Comptes

# CHAPITRE II – APPORTS

## APPORTS AU NIVEAU DES CONNAISSANCES TECHNIQUES

Ce projet nous a permis de :

* Savoir structurer les codes de sur le système MVC (Modèle Vue Controller)
* Savoir comment écrire un code épuré, portable, multi plateforme.
* Savoir comment utiliser les lectures sécurisées,
* Savoir les secrets de programmation sur les langages tel que : React.js, Java pour Android.
* Savoir comment bien utiliser l’outil GitHub pour le « versioning ».

## APPORTS AU NIVEAU DE LA CONCEPTION ET DU DÉVELOPPEMENT

Le projet nous a permis de :

* Maitriser les étapes de la conception au déploiement en passant par le développement.
* Savoir comment déléguer une tache et travailler en équipe.
* Apprendre à maitriser la méthodologie de conception et de développement des applications clientes.

## ÉVALUATION

### III.1) Bilan quantitatif

Pour notre programme on a :

* Nombre de vue : version mobile 25 Vues, version web 10 Vues
* Nombre de collection : 03 collections
* Nombre de relation : 03.

# CONCLUSION GÉNÉRALE

Pour conclure notre rapport de fin de stage. Où il était question pour nous de concevoir une application de gestion pour la microfinance SARKI MONEY CONTRIBUTION. En fin de les rendre encore plus compétitif sur le marché. Pour cela notre application va se remarquer un gain de temps considérable et une optimisation des taches. Car avec cette application les clients ; les collectrices et les administrateurs pourrons accédés aux informations en temps réel.

Pour les apport personnels notre stage a été très bénéfique dans la mesure où, il nous a permis de développé un esprit d’équipe, l’écoute et attention du détail car quand -il s’agit de l’argent une erreur peut couter très chère. Pour les apports techniques il nous a permis de mettre en pratique les connaissances vues en classe : le développement mobile sur le langage Java; le développement single page avec React ; la mise en place d’un backend No-SQL avec Firebase; la conception d’un système évolutif.

Certes notre application (SARKI-APP) est déjà fonctionnelle, suite au contrôle du maitre d’œuvre, ils en restent encore une fonctionnalité donc son développement faciliterais en plus la vie aux utilisateurs. Il s’agit de l’ajout de l’API des payements mobile money.

# WEBOGRAPHIE

# TABLE DE MATIÈRES

DÉDICACE i

REMERCIEMENTS ii

AVANT-PROPOS iii

SOMMAIRE iv

RÉSUME v

ABSTRACT vi

LISTE DES ABREVIATIONS vii

LISTE DES FIGURES viii

LISTE DES TABLEAUX ix

INTRODUCTION GÉNÉRALE 1

PARTIE 1 : RAPPORTE D’INSERTION 2

CHAPITRE 1 : PRESENTATION DE L’ENTREPRISE 3

I- PRESENTATION GENERALE DE LA STRUCTURE 3

I.1) Historique 3

I.2) Situation géographique 4

II- MISSIONS ET ORGANES DE GESTION 4

II.1) Services 4

II.2) Missions 4

II.3) Organes de gestion 5

CHAPITRE 2 : DÉROULEMENT DU STAGE 7

I- ACCUEIL 7

II- TÂCHES EFFECTUÉES 7

III- ÉVALUATION 7

III.1) Difficultés 7

III.2) Bilan de l’expérience 7

PARTIE 2 : CAHIER DES CHARGES 9

CHAPITRE 1 : DOSSIER D’ÉTUDE 10

I- CONTEXTE ET JUSTIFICATION DU PROJET 10

II- ÉTUDE DE L’EXISTANT 10

II.1) Présentation 10

II.2) Point fort 10

II.3) Point faible 10

II.4) Proposition de solution 11

III- LES OBJECTIFS DU PROJET 11

III.1) Objectif pour l’administration 11

III.2) Objectif pour le client 11

III.3) Objectif pour les collecteurs 11

IV- EXPRESSIONS DES BESOINS DE L’UTILISATEUR 12

IV.1) Besoin fonctionnel du projet 12

IV.2) Au niveau du client 12

IV.3) Au niveau du collecteur 12

IV.4) Au niveau de l’administrateur 12

IV.5) Besoin non fonctionnel du projet 12

IV.6) Fiabilité 12

IV.7) Ergonomie et bonne interface 13

IV.8) Sécurité 13

IV.9) Aptitude à la maintenance et la réutilisation 13

IV.10) L’intégrité 13

IV.11) Disponibilité : 24h/24 et 7j/7 BESOINS 13

V- ESTIMATION DU COÛT DU PROJET ET EQUIPE DE TRAVAIL 13

V.1) Équipe de travail 13

V.2) Resource matérielles 13

V.3) Ressources humaines 14

V.4) Grand total 14

VI- PLANIFICATION DU PROJET 14

VII- LES CONTRAINTES DU PROJET 15

CHAPITRE 2 : DOSSIER CONCEPTION 17

I- PRÉSENTATION DE LA MÉTHODE 2TUP 17

II- CONCEPTION 18

III- PRÉSENTATION DE UML 18

III.1) Diagramme fonctionnel 20

III.1.1) Diagramme des cas d’utilisation 20

III.1.2) Diagramme de cas d’utilisation du client 22

III.1.3) Diagramme de cas d’utilisation du collecteur 22

III.1.4) Diagramme de cas d’utilisation de l’administrateur 23

III.1.5) Diagramme de cas d’utilisation global 24

III.1.6) Description des cas d’utilisation 24

III.2) Modélisation conceptuelle des données 27

III.2.1) Dictionnaire des données 27

III.2.2) Diagramme de séquence 28

III.3) Quelques diagrammes de séquences du système 29

III.3.1) Collecte d’un client 29

III.3.2) Création d’un utilisateur 30

III.3.3) Authentification 31

III.4) Représentation des classes 31

CHAPITRE III : DOSSIER DE RÉALISATION 33

I- L’ÉTUDE TECHNIQUE 33

I.1) Environnement de réalisation 33

I.1.1) Matériels de base : 33

I.1.2) Choix des langages de développement et de SGBD : 34

I.1.3) Application web 35

I.1.4) Outil de développement 38

I.2) Modélisation physique des données 41

II- PRODUCTION DES PROGRAMMES 45

II.1) Descriptif du produit 45

II.1.1) Version mobile 45

II.1.2) Version web 45

II.2) Architecture 45

II.3) Charte graphique 45

II.3.1) Arrière-plan et texte 45

II.3.2) Usage des couleurs 46

II.3.3) Charte graphique stable et robuste 46

III- PHASE DE DÉPLOIEMENT 46

III.1) Diagramme de déploiement 46

III.1.1) Formalisme 46

III.1.2) Diagramme de déploiement 47

PARTIE III – BILAN 48

CHAPITRE 1 : PRÉSENTATION DE L’APPLICATION 49

I- LES VUES 49J

I.1) Version mobile 49

I.2) Version web 52

I.2.1) Les modèles 53

CHAPITRE II – APPORTS 55

I- APPORTS AU NIVEAU DES CONNAISSANCES TECHNIQUES 55

II- APPORTS AU NIVEAU DE LA CONCEPTION ET DU DÉVELOPPEMENT 55

III- ÉVALUATION 55

III.1) Bilan quantitatif 55

CONCLUSION GÉNÉRALE 56

WEBOGRAPHIE 56

TABLE DE MATIÈRES 56

ANNEXES 56

# ANNEXES