

**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**

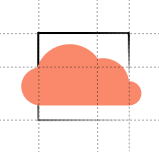
\*\*\*\*\*\*\*\* **INSTITUT UNIVERSITAIRE ET STRATEGIQUE DE L’ESTUAIRE**

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

**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 informatiques option Génie Logiciel**

**SOUS L’ENCADREMENT**

**Stage effectué du 27 Juin au 31 Septembre 2023 en vue de l’obtention du Diplôme de Technicien Supérieur (DTS)**

# DÉDICACE

**À mes parents**

# REMERCIEMENTS

# SOMMAIRE

DÉDICACE i

REMERCIEMENTS ii

SOMMAIRE iii

LISTE DES FIGURES iv

LISTE DES TABLEAUX v

GLOSSAIRE vi

RÉSUME vii

ABSTRACT viii

INTRODUCTION GÉNÉRALE 1

PARTIE 1 :PHASE D’INSERTION 2

CHAPITRE 1 : ACCEUIL EN ENTREPRISE 3

CHAPITRE 2 : PRÉSENTATION DE L'ENTREPRISE 7

PARTIE 2 : PHASE TECHNIQUE 9

DOSSIER 1 : L'EXISTANT 10

DOSSIER 2 : CAHIER DES CHARGES 17

DOSSIER 3 : DOSSIER D'ANALYSE 33

DOSSIER 4 : DOSSIER DE CONCEPTION 33

DOSSIER 5 : DOSSIER DE REALISATION 33

DOSSIER 6 : TESTS DE FONCTIONALITÉS 33

DOSSIER 7 : GUIDE D'INSTALLATION ET GUIDE UTILISATEUR 33

ANNEXE 1 : BIBLIOGRAPHIE 55

ANNEXE 2 : WEBOGRAPHIE 55

ANNEXE 3 : TABLE DES MATIERES 55

# 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

# GLOSSAIRE

# RÉSUMÉ

“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

# INTRODUCTION GÉNÉRALE

La présente introduction vise à présenter le rapport de stage réalisé dans le cadre du projet intitulé "Conception et réalisation d'une plateforme web et mobile d'achat de billets de transport inter-urbain". Ce projet s'inscrit dans un contexte où les besoins en matière de mobilité sont en constante évolution créant ainsi une certaine latence. La digitalisation des services de transport s’impose donc comme un enjeu majeur.

Le problème spécifique auquel nous nous sommes attaqués dans ce projet était de conçevoir une plateforme dédiée à l'achat de billets de transport inter-urbain. L'objectif était de créer une interface conviviale, intuitive et sécurisée, permettant aux utilisateurs de rechercher, réserver et acheter des billets de transport de manière efficace. Il était également crucial de prendre en compte les contraintes des heures, et des villes afin d'offrir une expérience utilisateur optimale.

Pour mener à bien ce projet, nous avons suivi un processus structuré comprenant plusieurs étapes. Dans un premier temps, nous avons procédé à une analyse approfondie des besoins et des attentes des utilisateurs ainsi que des contraintes techniques et opérationnelles. Ensuite, nous avons élaboré une stratégie de conception en définissant les fonctionnalités clés, l'architecture de la plateforme et les technologies à utiliser. Par la suite, nous avons entamé le développement de la plateforme en mettant en œuvre les différentes fonctionnalités et en assurant leur intégration harmonieuse. Enfin, nous avons réalisé des tests approfondis pour garantir la fiabilité, la sécurité et la performance de la plateforme, suivis d'une phase de déploiement et de suivi post-lancement.

Ce rapport de stage présentera en détail l'ensemble de ces étapes, en mettant l'accent sur les choix de conception, les défis rencontrés, les solutions développées, ainsi que les résultats obtenus.

# PARTIE 1 : RAPPORT D’INSERTION

# INTRODUCTION

Arrive en entreprise le 27 Juin 2023, avons été accueillie par le directeur général Johan Kenneth Nkeng, en qualité de stagiaire. Dans les lignes qui suivent nous présenterons dans un premier temps 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 | 12 Septembre 2016 |
| Siège social | Yaoundé Cameroun |
| Responsable | JOHAN KENNETH NKENG |
| Capital social | 556 000 000 FCFA |
| Page Facebook | SARKI Group |
| Slogan | « Servir bien pour, servir toute la vie » |

### I.2) Situation géographique

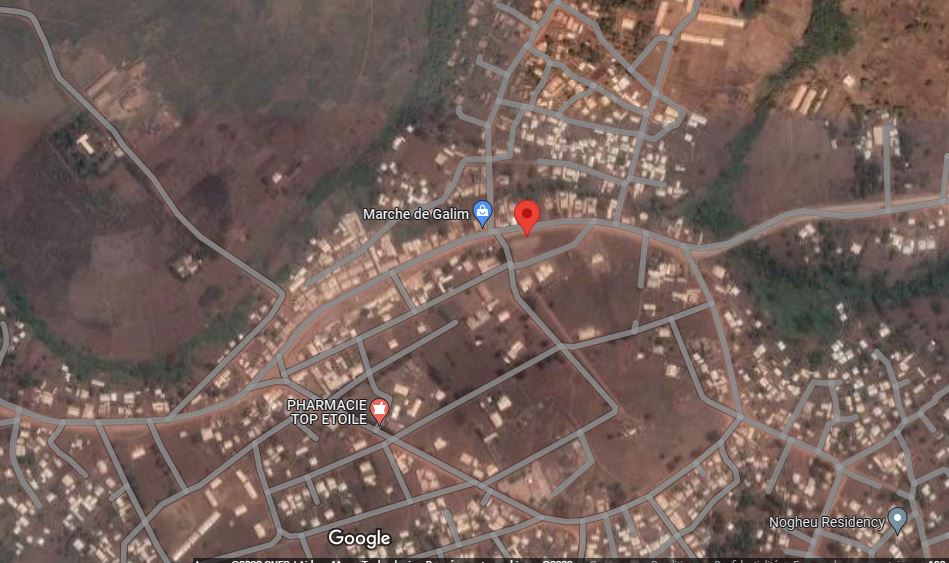


Figure 0: Situation géographique

## MISSIONS ET ORGANES DE GESTION

### II.1) Services

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

* Creation de sites web : 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

Mardi le 27 juin 2022 marque le début de notre stage à la startup Clouds Architects devant se poursuivre jusqu’au 30 Septembre 2022. Les premières semaines ont été marquées par plusieurs étapes parmi lesquelles :

* la prise en contact avec les membres du personnel
* La présentation du fonctionnement interne de l’entreprise et des technologies utilisées !
* La présentation des différentes contraintes de l’entreprise notamment les heures d’arrivée, de départ, de pause, des meetings en début et en fin de journée.
* L’inscription sur la plateforme Trailhead, dédiée à l’apprentissage des outils et technologies Salesforce .

L’on nous a également exhortés à nous mettre au travail dès le début afin de garantir un stage instructif et constructif !

## TÂCHES EFFECTUÉES

Nous avons au sein de Clouds Architects effectué plusieurs taches durant notre stage, aux rangs desquelles :

* + Réalisation de défis et résolution de problèmes sur la plateforme Trailhead.
  + Installation d’utilitaire sur des postes défectueux,
  + Participation à la conception de sites web,
  + Aménagement de mon espace de travail

## ÉVALUATION

### III.1) Difficultés

Le plus difficile lors de notre stage était les soucis rencontrés avec les coupures récurrentes d’électricité incluant une perte de connexion à internet.

### III.2) Bilan de l’expérience

Tout d’abord, il est important de noter que notre stage s’est déroulé dans des bonnes conditions. Grâce à une équipe dynamique, compétente et surtout sociable, notre intégration dans le groupe s’est faite non pas sans difficultés mais sans problème majeur.

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 employés de la structure.
* Le développement de l’écoute; capitale en entreprise afin de produire un résultat fidèle aux souhaits des clients, et aux préoccupations de nos collègues.
* L’humilité de demander de l’aide quand cela est nécessaire et la facilité de porter assistance au mieux de nos compétences à tous les utilisateurs dans le besoin.

# PARTIE 2 : PHASE TECHNIQUE

# 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.

# DOSSIER 1 : L’EXISTANT

## PRÉSENTATION DU THÈME

Dans notre quotidien, que ce soit pour nous rendre à un évènement particulier, pour des vacances ou même pour du tourisme, nous sommes appelés à nous déplacer d’une ville à l’autre. Au Cameroun, pour se faire, l’on se sert principalement d’agences de transport interurbain. Toutefois, le système mis en place nous semble avoir un certain nombre de lacunes qui impliquent une perte de temps, et éventuellement d’argent. Afin de rendre plus autonome le système, nous avons donc jugé nécessaire et surtout optimal de traiter notre thème : mise en place d’une plateforme web et mobile de réservation de billets de transport urbain. Une plateforme qui regroupe les différents voyages créés par les agences de voyages ; accessible aux clients qui après une inscription peuvent rechercher, réserver et acheter un billet pour un voyage.

## ÉTUDE DE L’EXISTANT

On peut procéder de plusieurs façons pour décrire le système actuel. Mais, que ce soit en interrogeant les clients, les fournisseurs ou même en nous rendant sur place, on retrouve à peu près les mêmes réponses.

Dans l’existant, les agences sont dissociées les unes des autres et utilisent des systèmes isolés, et centralisés associés à un générateur de facture impliquant donc une assez grande dépendance au papier.

Du côté du client, nous pouvons dire que le système n’est pas vraiment à l’avantage du système :

-Pour toute information ou renseignements que ce soit pour la grille de prix ou les horaires, le client doit se rendre à l’agence qui l’intéresse.

- Les différentes agences possèdent généralement une boîte à suggestion.

- La réservation se fait sur place ou au téléphone suite à une interaction avec un agent de l’agence concernée.

-Chaque agence possède son propre système ; le client doit donc faire le tour si il souhaite faire des comparaisons de services.

## CRITIQUE DE L’EXISTANT

Le système de transport interurbain présent dans notre société est fonctionnel, et on peut y observer des points forts mains aussi des points faibles.

### II.2) Point fort

* La présence d’une traçabilité des données et des différentes transactions.
* La collaboration et la présence de la confiance entre les différents acteurs (clients, agence) dues à la proximité.

### II.3) Point faible

* Absence presque systématique d’une base de données relationnelle.
* Risque de Pertes des données : En général, les données ne sont pas stockées sur le web.
* Le client doit forcément se déplacer ou appeler pour réserver et éventuellement arriver plus tôt pour s’assurer d’une place ou s’enquérir des prix de chaque agence.
* Des problèmes récurent de monnaie.
* Le client ne peut pas comparer les prix en fonction des destinations sans se rendre dans les différents sièges des différentes entreprises.

Nous observons donc un système fonctionnel mais avec des inconvénients. Notre système viendra résoudre les points faibles évoqués et rendre le système de réservation de billet plus performant ainsi que celui de gestion de billets.

## PROBLEMATIQUE

La problématique du sujet "Conception et réalisation d'une application web et mobile de réservation de billets de bus de transport interurbain" réside dans la nécessité de répondre aux besoins croissants des utilisateurs en matière de mobilité et de faciliter l'accès aux services de transport interurbain. Dans un contexte où les déplacements entre les villes sont de plus en plus fréquents, il est essentiel de proposer une solution pratique et conviviale pour la réservation de billets de bus.

Cependant, les systèmes de réservation traditionnels peuvent présenter des limitations, tels que la nécessité de se rendre physiquement à une agence ou de réserver par téléphone, ce qui peut être contraignant et chronophage pour les utilisateurs. De plus, l'absence d'une plateforme centralisée et facilement accessible peut rendre difficile la comparaison des horaires, des tarifs et des disponibilités entre les différents transporteurs.

En résolvant cette problématique, le projet vise à améliorer l'expérience des utilisateurs en leur offrant une solution pratique et efficace pour la réservation de billets de bus interurbains, tout en favorisant la digitalisation du secteur des transports. Il s'agit également de faciliter la gestion des réservations pour les transporteurs, en automatisant les processus et en réduisant les erreurs liées à la réservation manuelle.

En conclusion, la problématique de ce projet est d'apporter une solution innovante qui réponde aux besoins des utilisateurs en matière de réservation de billets de bus interurbains, tout en facilitant l'accès à l'information, en améliorant l'efficacité des processus et en offrant une expérience utilisateur optimale.

## PROPOSITION DE SOLUTION

Pour Pallier aux problèmes cités plus haut, notre système sera mis en avant garantissant un une amélioration des services déjà proposés. Elle pourra aussi assurer une plus grande autonomie au système, permettra aux clients de réserver de n’importe où, avec son téléphone mobile Android ou un ordinateur en lui donnant accès aux voyages prévus par ses agences favorites ; assurera à l’administration des outils pour une meilleure organisation de leurs différentes ressources et donc des différents voyages. Le tout, accessible en temps réel de manière sécurisée.

# DOSSIER 2 : CAHIER DES CHARGES

## CONTEXTE ET JUSTIFICATION DU PROJET

## LES OBJECTIFS DU PROJET

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

### Le projet pourra aider les administrations des différentes agences de voyage à optimiser la surveillance des différents voyages, de leurs différentes ressources (chauffeurs, bus etc).

### III.2) Objectifs pour le client

Le projet a pour objectif principal de permettre aux clients de visualiser les différents voyages disponibles, de comparer les différentes agences correspondantes, et réserver son billets de n’importe où.

## EXPRESSIONS DES BESOINS DE L’UTILISATEUR

### III.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 :

### III.2) Au niveau du client

* S’inscrire
* Se connecter
* Se déconnecter
* Voir le listing des dernières réservations / derniers voyages.
* Écrire au support d’une agence ou de la plateforme.
* Effectuer une réservation en se basant sur les voyages disponibles.

### IV.3) Au niveau de l’administrateur

* Se connecter
* Se déconnecter
* Gérer (ajout, modification; suppression) les ressources de l’agence (chauffeurs, bus, motoboys) et les voyages.
* Pourvoir ouvrir un compte pour un client.
* Faire une réservation pour un client.
* Consulter l’historique des voyages et les informations relatives.

### III.4) Au niveau du super-administrateur

* Gérer (ajout, modification, suppression) les administrateurs, les agences, les dispositions.
* Consulter les messages reçus.
* Voir toutes les transactions du système

### III.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.

### III.6) Fiabilité

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

### III.7) Ergonomie et l’utilisabilité

### Les efforts fournis par l’utilisateur pour l’utilisation satisfaisante de l’application doivent être minimes.

### III.8) Sécurité

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

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

### III.10) L’intégrité

Les données enregistrées ne doivent pas pouvoir être modifiées par une tierce personne sans autorisation ni être perdus.

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

## 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** | **Juillet** | | | | | **Août** | | | | **Septembre** | | | | |
|  | **1** | **2** | **3** | **4** | **1** | | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **Étude préalable** |  |  |  |  |  | |  |  |  |  |  |  |  |
| **Conception** |  |  |  |  |  | |  |  |  |  |  |  |  |
| **Réalisation** |  |  |  |  |  | |  |  |  |  |  |  |  |
| **Test et validation** |  |  |  |  |  | |  |  |  |  |  |  |  |
| **Rédaction du rapport** |  |  |  |  |  | |  |  |  |  |  |  |  |

### 

## 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** |
| Fogue tiagho arsene nelson | Modélisation, Conception, et Réalisation |
| M. FENKAM Jules Blaise | Encadreur académique et chargé de communication à l’IAI Cameroun Centre technologique d'excellence de Paul Biya |
| M. NKENG Kenneth Johan | Chef de projet. Encadreur professionnel. Directeur de CLOUDS ARCHITECTS. |

### V.2) Resource matérielles

Tableau 02 : ressource matérielle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Matériels** | **Fonctions** | **Prix unitaire** | **Quantité** | **Prix total (FCFA)** |
| **Ordinateur : i7 2e Gen 6Go Ram** | Équipement de développement | **150 000** | **03** | **450 000** |
| **Téléphone Android : 9, 2Go ram** | Équipement de test | **60 000** | **02** | **120 000** |
| **Modem Huawei** | Accès à internet | **150 000** | **01** | **150 000** |
| Total | **720 000 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** | 1 | 200 000 |
| **Tester** | 15 | 300 000 |
| **Total** | 15 | 1 000 000 |

### V.4) Grand total

Tableau 04 : Total

|  |  |
| --- | --- |
| **Types** | **Prix (F CFA)** |
| **Matérielles** | **720 000** |
| Humaines | **1 000 000** |
| Imprévues | **1 720 000 \* 15 %** |
| Total de cout | **1 978 000** |

## 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 | il s'agit du calendrier de livraison du projet, notamment les dates de livraison de chaque phase du projet, ainsi que la date de livraison du livrable final. Ne pas respecter les deadlines des différentes phases peut entrainer un retard du projet global. Étant limité dans la réalisation de notre projet (03) mois; une planification optimale est nécessaire. |
| 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 au 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.

# DOSSIER 3 : DOSSIER D’ANALYSE

## METHODOLOGIE

## I.1) ETUDE COMPARATIVE UML ET MERISE

MERISE ET UML sont des methodes/ langages très utilisés dans les projets de developement logiciel , Choisir lequel choisir est un une étape crutiale de notre projet, il est donc important d’analyser les contours de notre projet et de connaitre les différences entre les deux méthodes avant de faire le choix. Les différences entre UML et MERISE peuvent s’observer à plusieurs niveaux.

1. Introduction :
   * UML : L’unified modeling language est un langage graphique standard utilisé pour modéliser des systèmes logiciels. Il fournit une notation visuelle pour représenter des concepts et des relations dans un système.
   * Merise : Merise (**Méthode d'étude et de réalisation informatique pour les systèmes d'entreprise)** est une méthode de modélisation conceptuelle utilisée pour concevoir des systèmes d'information. Elle se concentre sur la modélisation des données, des processus métier et des interactions entre eux.
2. Domaines d'application :
   * UML : UML est principalement utilisé dans le domaine du développement logiciel. Il permet de modéliser des systèmes logiciels complexes, tels que des applications, des systèmes embarqués ou des architectures orientées services.
   * Merise : Merise est principalement utilisé dans le domaine de la gestion des systèmes d'information. Il est utilisé pour concevoir et modéliser des bases de données, des schémas de flux de données et des processus métier.
3. Niveaux de modélisation :
   * UML : UML offre une variété de diagrammes qui permettent de modéliser différents aspects d'un système. Par exemple, le diagramme de cas d'utilisation représente les interactions entre les acteurs et le système, tandis que le diagramme de classes montre la structure des classes et leurs relations.
   * Merise : Merise propose des modèles conceptuels, logiques et physiques pour la modélisation des données, des processus et des interactions. Le modèle conceptuel des données (MCD) représente les entités, les relations et les contraintes, tandis que le modèle logique des données (MLD) définit la structure détaillée de la base de données.
4. Approche de modélisation :
   * UML : UML adopte une approche orientée objet, mettant l'accent sur la représentation des classes, des objets, des relations et des comportements. Il permet de modéliser des systèmes en identifiant des abstractions et en décrivant comment ces abstractions interagissent.
   * Merise : Merise adopte une approche entité-association, mettant l'accent sur la représentation des entités, des attributs, des relations et des processus métier. Il se concentre sur la modélisation des données et des flux d'informations dans un système.
5. Notations et diagrammes :
   * UML : UML utilise différentes notations graphiques pour représenter les concepts et les relations. Par exemple, les classes sont représentées par des rectangles avec des attributs et des méthodes, les relations sont représentées par des flèches entre les classes, et les interactions sont représentées par des séquences de messages.
   * Merise : Merise utilise des notations graphiques spécifiques pour représenter les entités, les relations, les processus et les flux de données. Par exemple, les entités sont représentées par des rectangles, les relations sont représentées par des flèches, les processus sont représentés par des ovales, et les flux de données sont représentés par des flèches entre les processus.
6. Focus :
   * UML : UML met l'accent sur la modélisation des aspects structurels et comportementaux des systèmes logiciels. Il permet de représenter les classes, les objets, les relations, les états, les événements et les interactions entre les composants d'un système.
   * Merise : Merise met l'accent sur la modélisation des aspects structurels et fonctionnels des systèmes d'information. Il permet de représenter les données, les processus métier, les interactions entre les acteurs et les flux d'informations.
7. Utilisation complémentaire :
   * UML : UML est souvent utilisé en conjonction avec des méthodologies de développement logiciel telles que les méthodes agiles ou en cascade. Il permet de documenter et de communiquer les différentes phases du cycle de vie d'un projet logiciel.
   * Merise : Merise est souvent utilisé avec d'autres méthodes de gestion des systèmes d'information, telles que ITIL ou COBIT. Il permet de concevoir et de modéliser les systèmes d'information en se concentrant sur les données, les processus et les interactions.
8. Popularité et adoption :
   * UML : UML est largement adopté dans l'industrie du développement logiciel. Il est soutenu par de nombreux outils de modélisation et est enseigné dans de nombreux programmes d'informatique.
   * Merise : Merise est populaire en France et dans les pays francophones, où il est largement utilisé dans le domaine de la gestion des systèmes d'information. Il bénéficie d'une adoption significative dans les organisations qui suivent des normes de gestion spécifiques.
9. Flexibilité et extensibilité :
   * UML : UML offre une grande flexibilité et la possibilité d'étendre le langage en définissant des profils spécifiques à un domaine ou à une entreprise. Cela permet d'adapter UML aux besoins particuliers d'un projet ou d'une organisation.
   * Merise : Merise est moins flexible et moins extensible que UML, car il est plus spécifiquement axé sur la modélisation des systèmes d'information. Il offre cependant des concepts et des méthodes éprouvés pour la conception des systèmes d'information.
10. Choix de méthode :
    * UML : UML est préféré pour la modélisation des systèmes logiciels complexes ou orientés objet. Il est largement utilisé dans le développement d'applications, de systèmes embarqués et de systèmes distribués.
    * Merise : Merise est préféré pour la modélisation des systèmes d'information, en particulier dans les projets où la gestion des données et des processus est primordiale. Il est couramment utilisé pour concevoir des bases de données et des systèmes d'information dans divers secteurs d'activité.

## I.1) ETUDE COMPARATIVE DES PROCESSUS UNIFIÉS

## II- MODELISATION

## II.1) DIAGRAMME DE CAS D’UTILISATIONS

Le diagramme de cas d’utilisation (Use Case Diagram) constitue la première phase de l’analyse UML ; elle se concentre sur un certain nombre de tâches telles que :

* La modélisation les besoins des utilisateurs.
* L’identification les grandes fonctionnalités et les limites du système.
* La représentation des interactions entre le système et ses utilisateurs.
* La composition 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 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. De même 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 achète son ticket.
* L’administrateur: Il utilise le système pour administrer les ressources de son agence et effectue une réservation pour un client.
* Le super-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** |
| Acheter un ticket | Client |
| Consulter la de ses tickets achetés | Client |
| Gérer les bus, les voyages, les chauffeurs, les réservations, les motoboys. | Administrateur |
| Consulter l’historique des voyages de son agence. | Administrateur |
| Gérer les admins, les agences,les dispositions. | Super-Administrateur |
| Faire un listing des voyages dispongnibles | Client,administrateur |

## II.1.1) Diagramme de cas d’utilisation du client

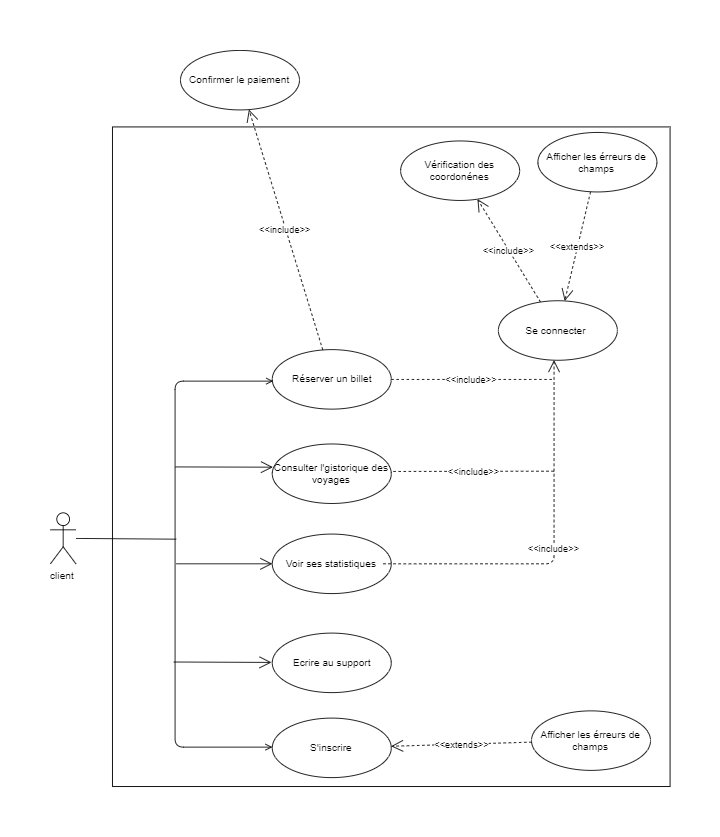


Figure 01 : cas d’utilisation du client

#### III.1.2) Diagramme de cas d’utilisation de l’administrateur

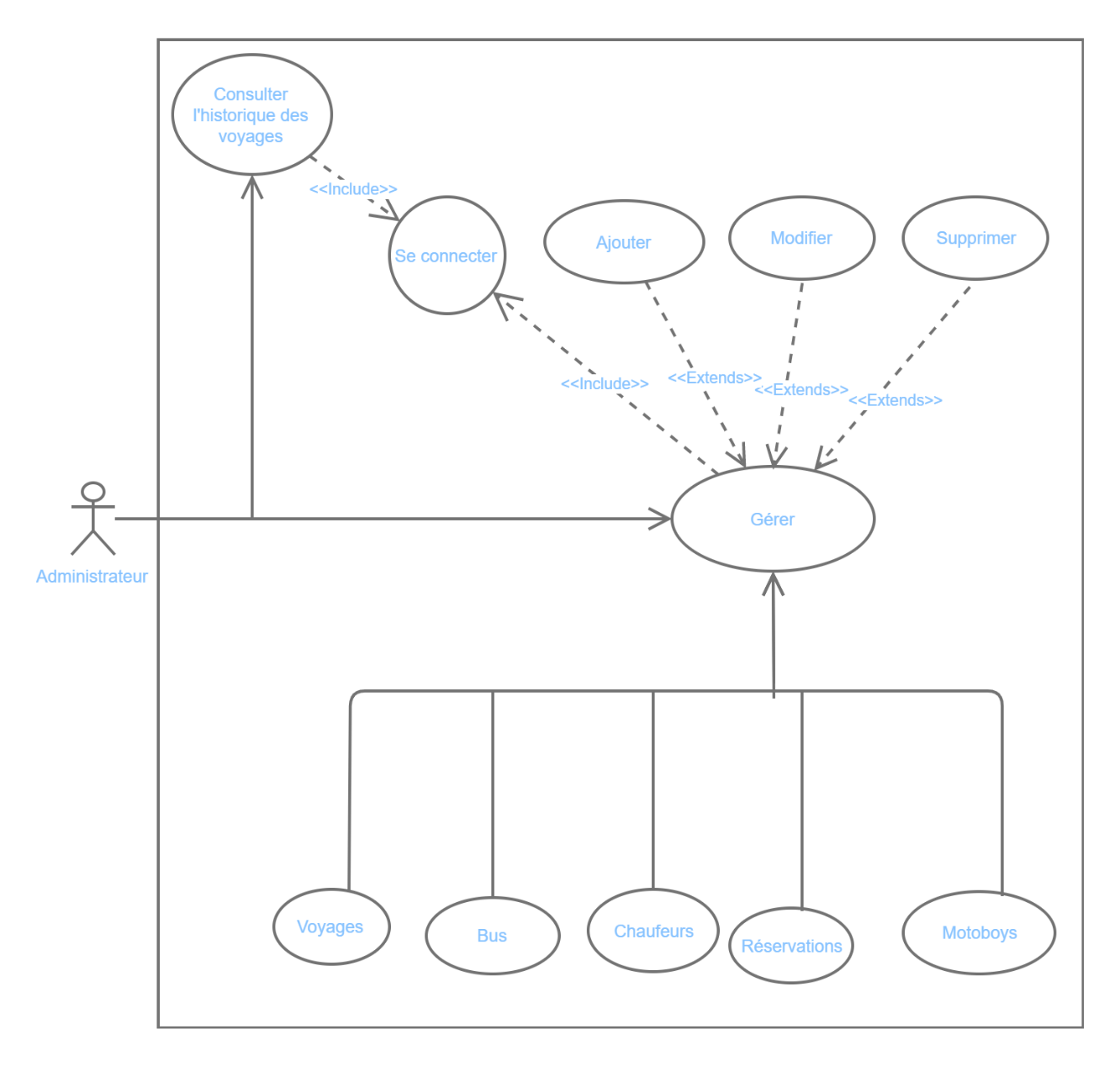


Figure 02 : cas d’utilisation de l’administrateur

#### III.1.4) Diagramme de cas d’utilisation du super-administrateur

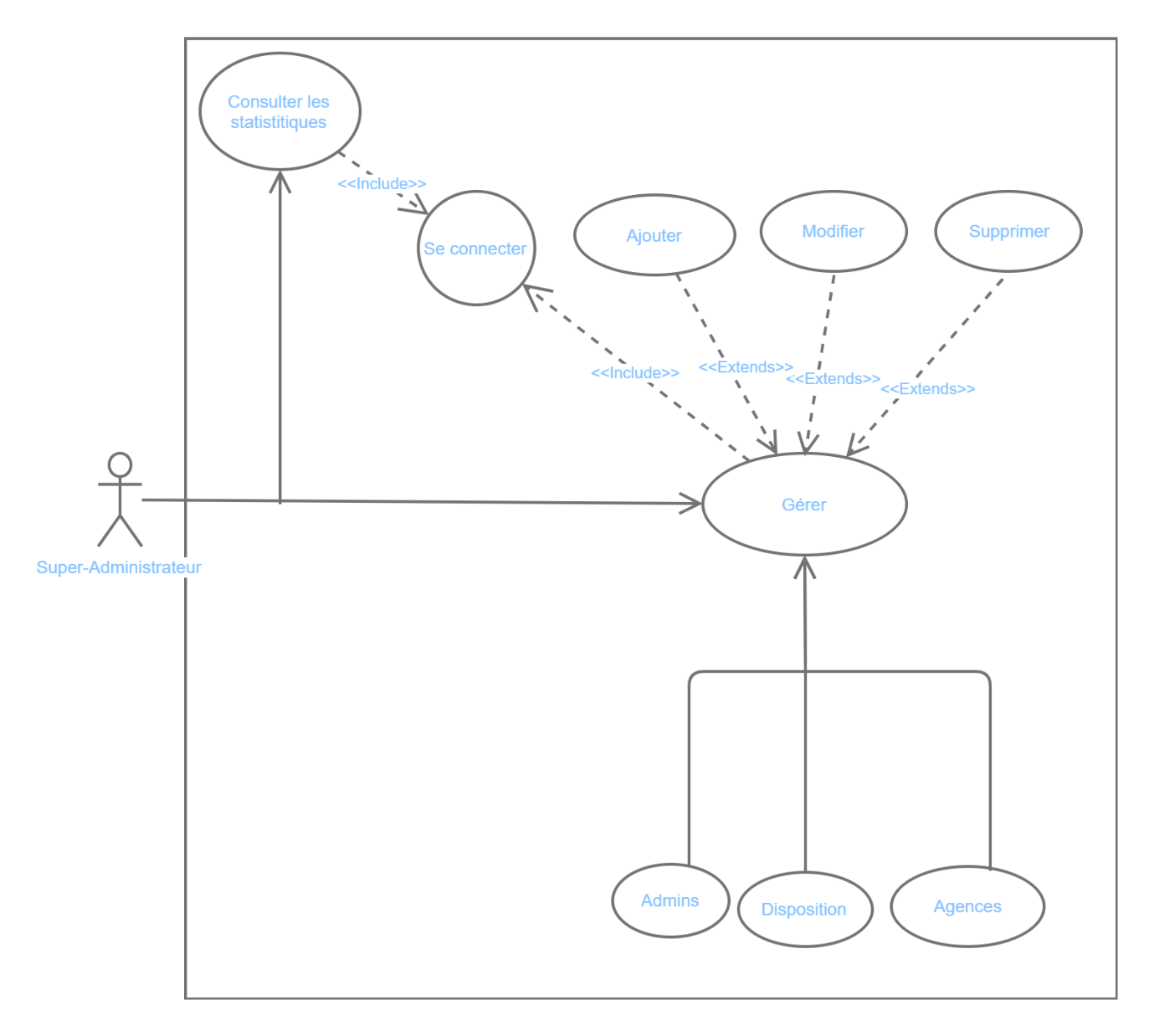


Figure 03 : cas d’utilisation du super-administrateur

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

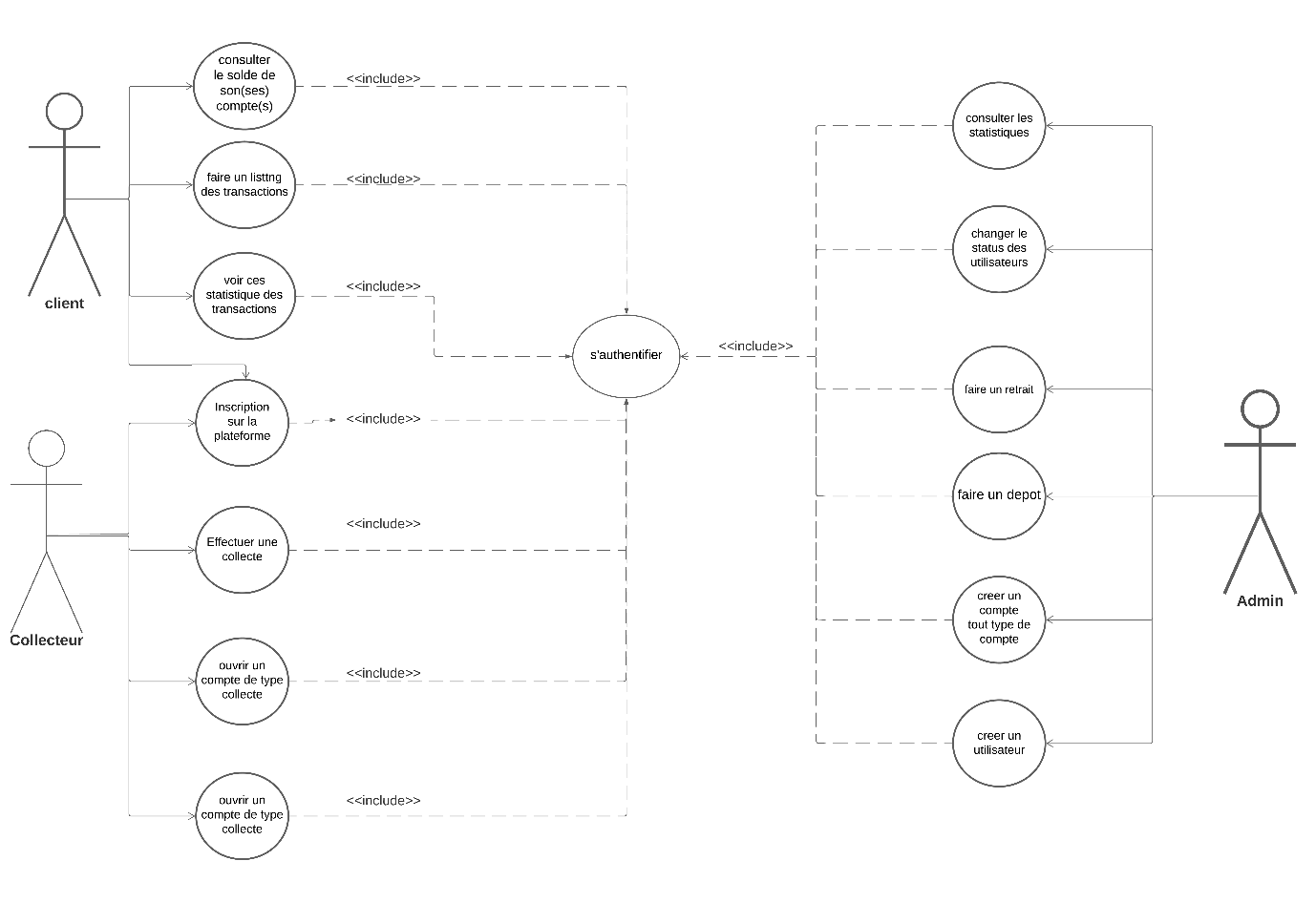


Figure 04 : cas d’utilisation Global

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

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

Tableau 09 : cas d’utilisation d’authentification

|  |
| --- |
| CU1 : Authentification |
| Résumé : permet aux différents acteurs d’accéder à leur espace de travail |
| Acteurs : administrateur, client, super-administrateur |
| Scénario nominal |
| DESCRIPTION DU SCENARIO NOMINAL  « Début »  01 : saisir les informations d’authentification  02 : cliquer sur login  « Fin » « Fin » |
| DESCRIPTION DU SCENARIO ALTERNATIF |
| Les informations entrées ne sont pas valides : Coordonnées invalides. |

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

|  |
| --- |
| CU2 : Créer un administrateur |
| Résumé : L’on crée un administrateur pour une agence précédemment enregistré, afin qu’il puisse Gérer les ressources de son agence. |
| Acteur : Super-administrateur |
| Scénario nominal |
| DESCRIPTION DU SCENARIO NOMINAL  « Début »  01 : authentification  02 : saisir les informations de l’administrateur  03 : confirmer la création  « Fin » |
| Scénario alternative |
| DESCRIPTION DU SCENARIO ALTERNATIF  Les informations entrées ne sont pas valides : Une erreur est renvoyée au super-administrateur en fonction du champ mal renseigné. |
|  |

Tableau 11 : cas d’utilisation de réservation d’un ticket

|  |
| --- |
| CU3 : Réserver un ticket |
| Résumé : Permettre à l’utilisateur de faire un choix sur le voyage qu’il souhaite effectuer et de réserver son ticket. |
| Acteur : client |
| Scénario nominal |
| DESCRIPTION DU SCENARIO NOMINAL  « Début »  01 : se connecter  02 : Cliquer sur Faire une réservation  03 : Entrer les informations sur le voyage que l’on recherche (date, heure, ville de départ, ville d’arrivée).  04 : Cliquer sur Réserver, sur la carte du voyage qui nous intéresse.  05 : Entrer les informations du voyageur et de paiement, puis sur Réserver.  06 : Valider la requête de retrait reçu sur notre téléphone.  « Fin » |
| Scénario alternatif |
| DESCRIPTION DU SCENARIO ALTERNATIF  -Les informations entrées ne sont pas valides : Renvoi d’un message d’erreur correspondant au champ mal renseigné.  -L’utilisateur ne valide pas la demande de retrait : Message d’erreur « la requête a été annulée, veuillez réessayer s’il vous plait ! » |
|  |

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

|  |
| --- |
| CU4 : Ajouter un voyage |
| Résumé : permettre de créer un nouveau voyage |
| Acteur : administrateur |
| Scénario nominal |
| DESCRIPTION DU SCENARIO NOMINAL  « Début »  01 : se connecter  02 : cliquer sur voyage  03 : Entrer les informations du voyage telles que le code, le bus, le chauffeur, le motoboy, l’heure de départ, les villes de départ et d’arrivée.  04 : Cliquer sur Ajouter.  « Fin » |
| Scénario alternative |
| DESCRIPTION DU SCENARIO ALTERNATIF  Les informations entrées ne correspondent pas au format demandé: Renvoi d’un message d’erreur correspondant au champ mal renseigné. |

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. |

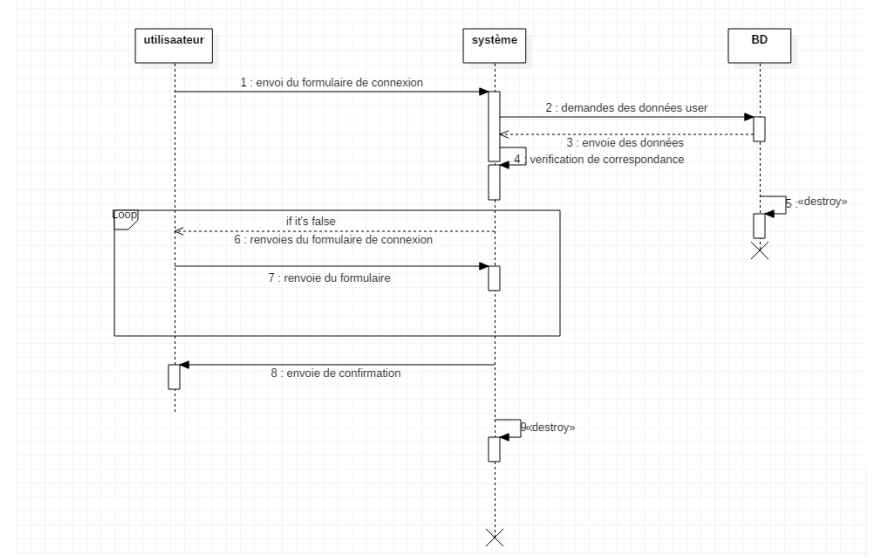
## II.2) DIAGRAMME DE SEQUENCES

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.

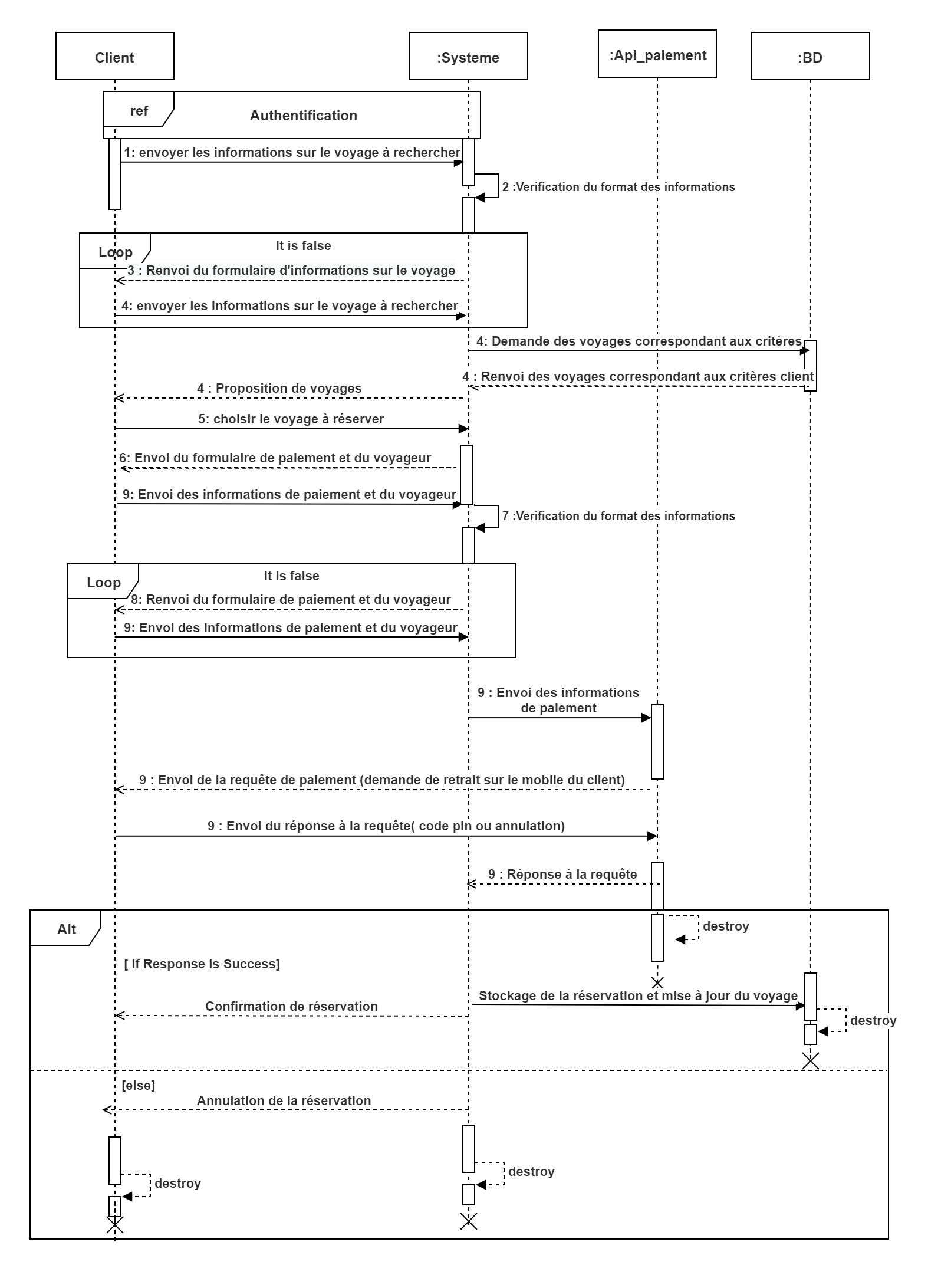
Tableau 15 : formalisme du diagramme de séquences

|  |  |  |
| --- | --- | --- |
| **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** | |  |  | | --- | --- | |  | Représente le temps nécessaire pour qu'un objet accomplisse une tâche. Plus la tâche nécessite de temps, plus la boîte d'activation est longue. | |  |

#### III.2.1) Diagramme de séquence : Authentification



#### III.2.2) Diagramme de séquence : Réservation



## II.3) DIAGRAMME D’ACTIVITÉS

Tableau 16 : formalisme du diagramme d’activités

|  |  |  |
| --- | --- | --- |
| **Composant** | **Description** | **Symbole** |
| **Symbole de début** | Représente le début d'un processus ou d'un flux de travail dans un diagramme d'activités. Il peut être utilisé seul ou avec un symbole de note qui explique le point de départ. | symbole de début |
| **Symbole d'activité** | Indique les activités qui composent un processus modélisé. Ces symboles, qui comprennent de brèves descriptions dans la forme, sont les principales composantes d’un diagramme d’activités. | symbole d'activité |
| **Symbole de raccord** | Indique le flux directionnel, ou flux de contrôle, de l'activité. Une flèche entrante marque le début d'une étape d'une activité ; une fois l'étape terminée, le flux se poursuit avec la flèche sortante. | Symbole de raccord |
| **Symbole de raccord/barre de synchronisation** | Associe deux activités simultanées et les réintroduit dans un flux où n'a lieu qu'une seule activité à la fois. Représenté par une ligne verticale ou horizontale épaisse | symbole de raccord |
| **Symbole d'embranchement** | Divise un flux d'activités en deux activités simultanées. Symbolisé par plusieurs lignes fléchées qui partent d'un raccord. | symbole d'embranchement |
| **Remarque** | Permet aux créateurs d'un diagramme ou à leurs collaborateurs de communiquer des messages supplémentaires qui n'entrent pas dans le diagramme à proprement parler. Permet de laisser des notes pour plus de clarté et de précision. | Note |
| L'**activité de décision** | L'**activité de décision** est introduite dans UML pour supporter les conditionnels dans les activités. Une activité de décision est modélisée comme un diamant sur un diagramme d'activité UML. L'activité de décision doit refléter l'activité précédente. | Activité de décision |
| **Symbole de fin** | Marque l’état final d’une activité et représente l’achèvement de tous les flux d’un procédé. | symbole de fin |

# DOSSIER IV : DOSSIER DE CONCEPTION

### Diagramme 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.

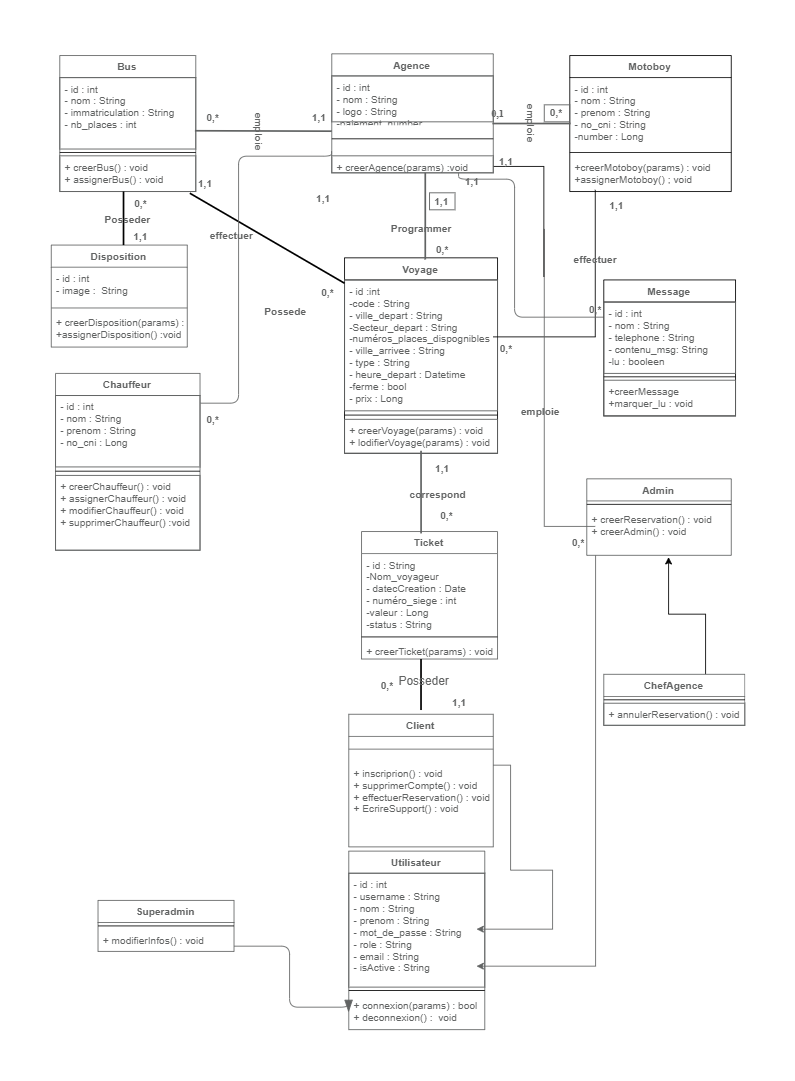


Figure 08 : diagramme de Classe

### 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

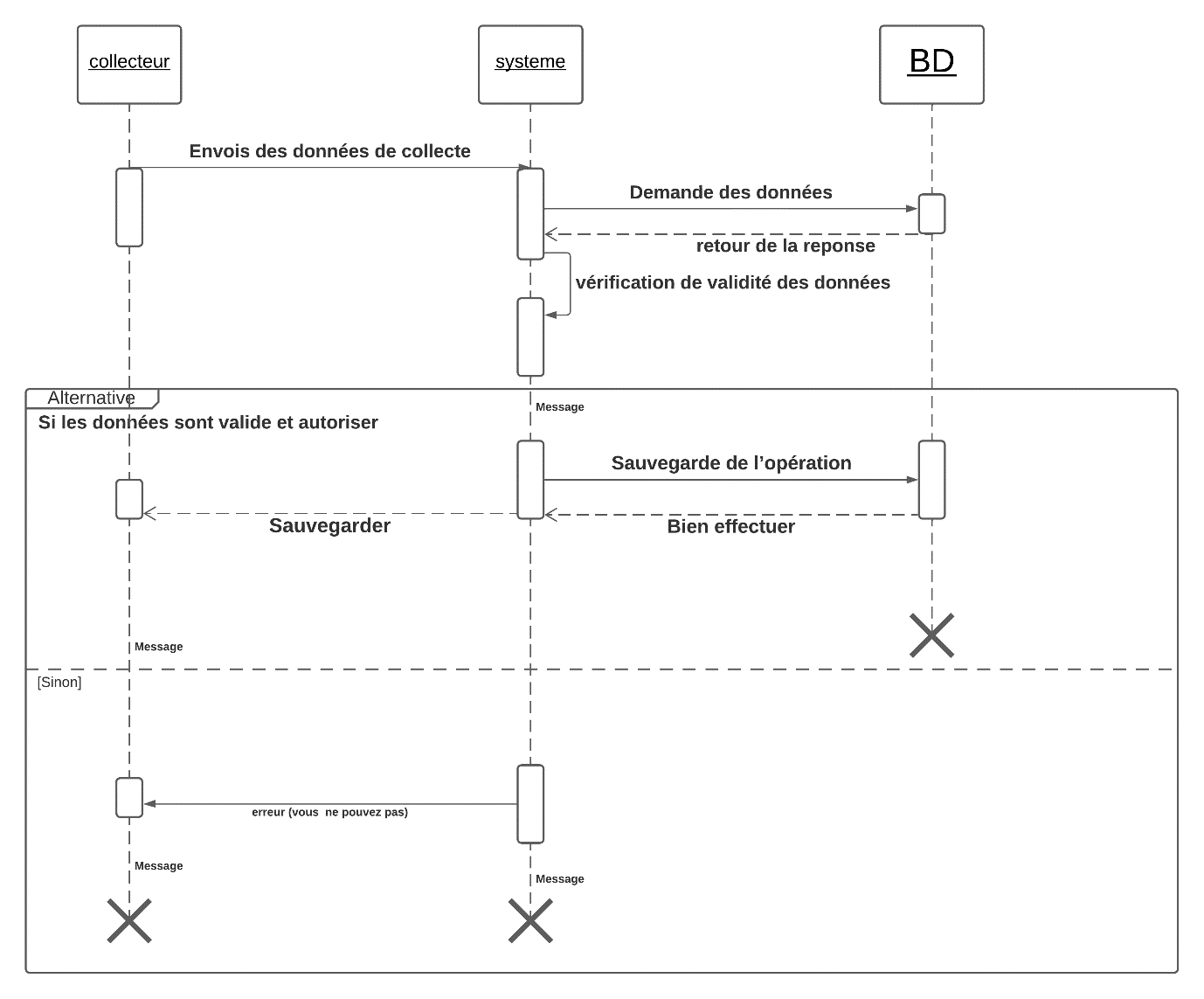


Figure 05 : diagramme de séquence d’une collecte

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

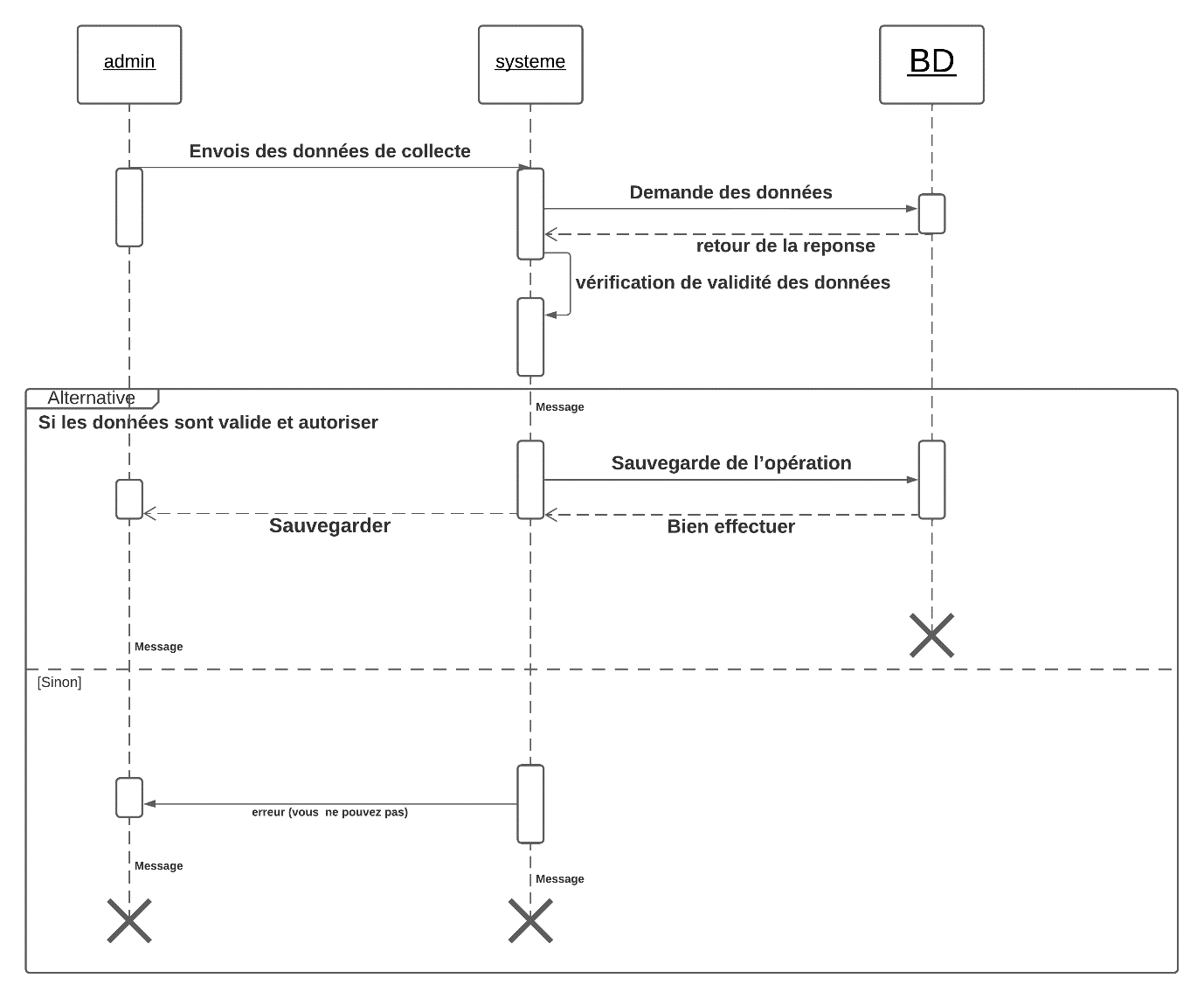


Figure 06 : diagramme de séquence de la création d’un utilisateur par l’administrateur

#### III.3.3) Authentification

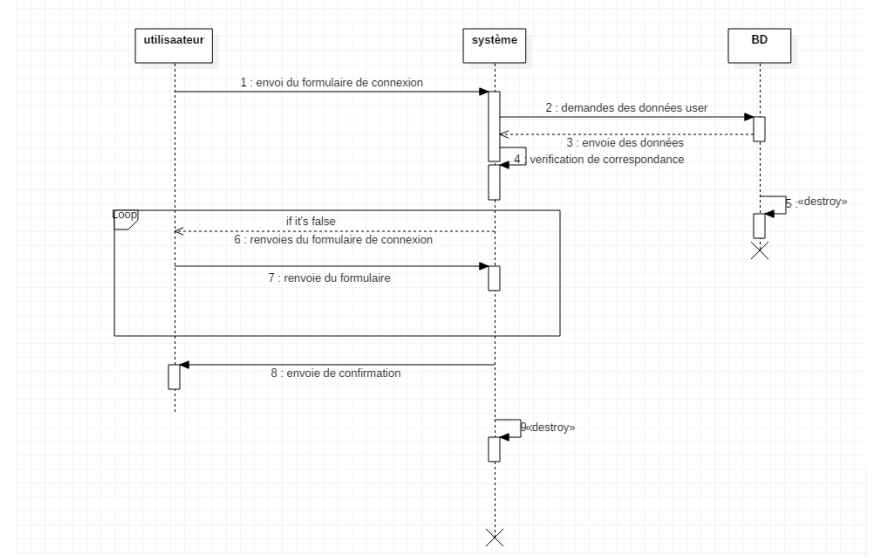


Figure 07 : diagramme de séquence d’authentification

# 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" \o "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

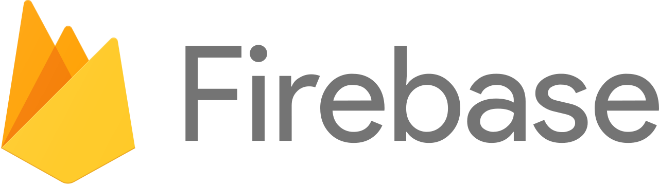


Figure 11 : logo Firebase

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)" \o "IOS (Apple)), [Javascript](https://fr.wikipedia.org/wiki/Javascript" \o "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)" \o "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" \o "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" \o "Twitter) et [Github](https://fr.wikipedia.org/wiki/Github" \o "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

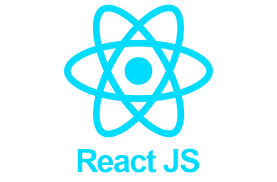


Figure 12 : logo React JS

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" \o "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**

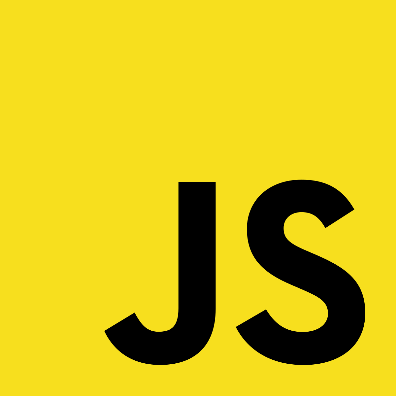


Figure 13 : logo JavaScript

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" \o "API ɡ)[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" \o "en: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" \o "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" \o "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 »

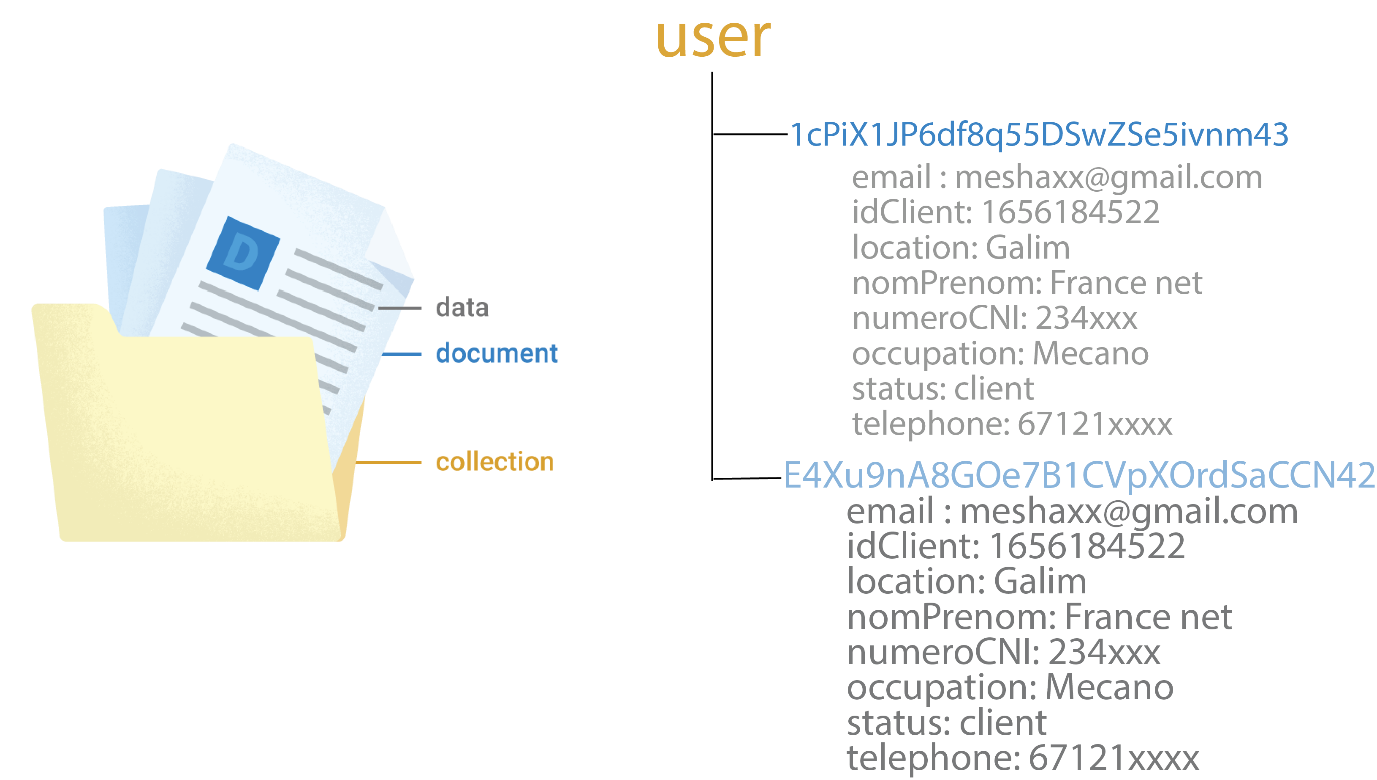


Figure 21 : représentation de la collection des utilisateur

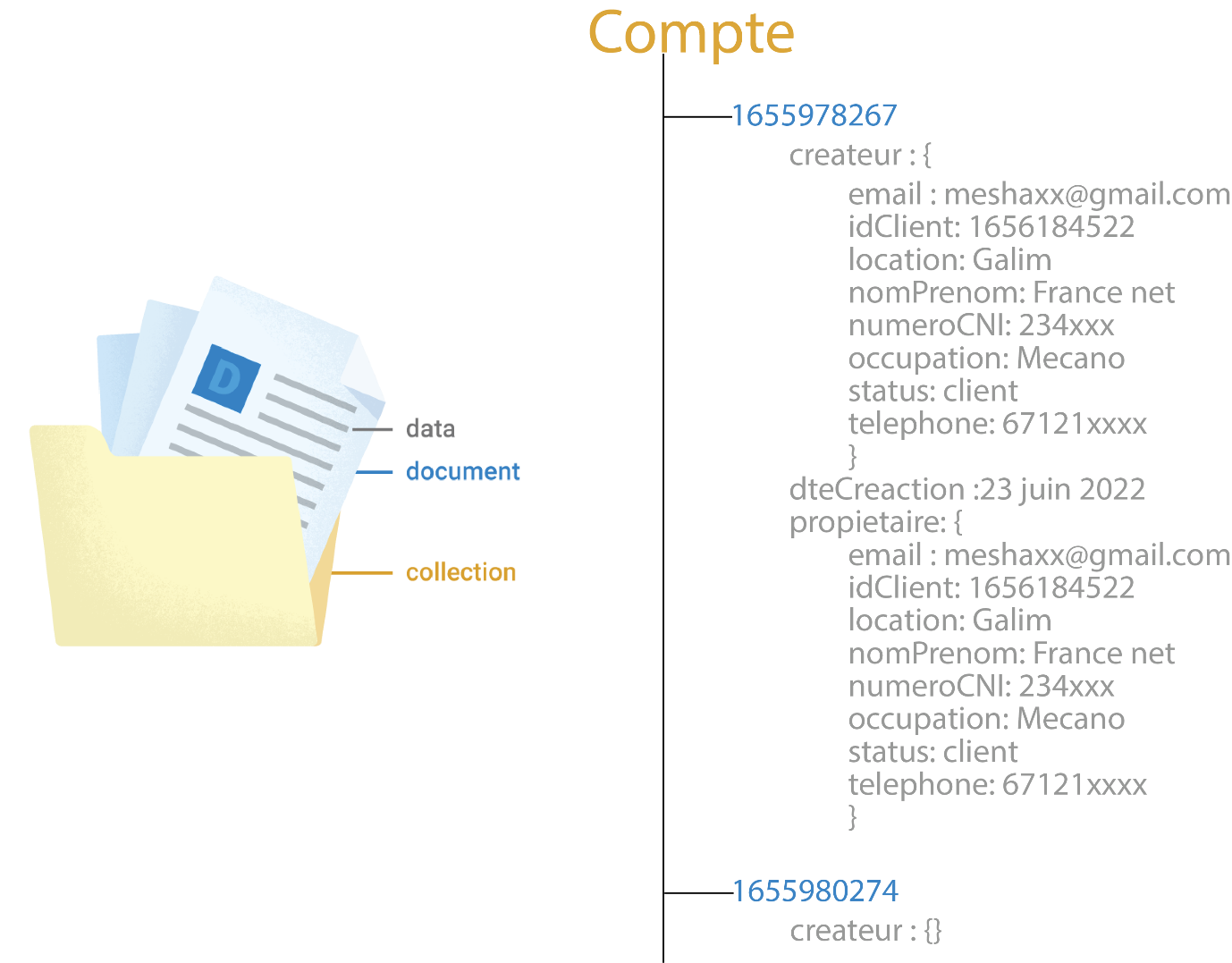


Figure 22 : représentation de la collection des Comptes

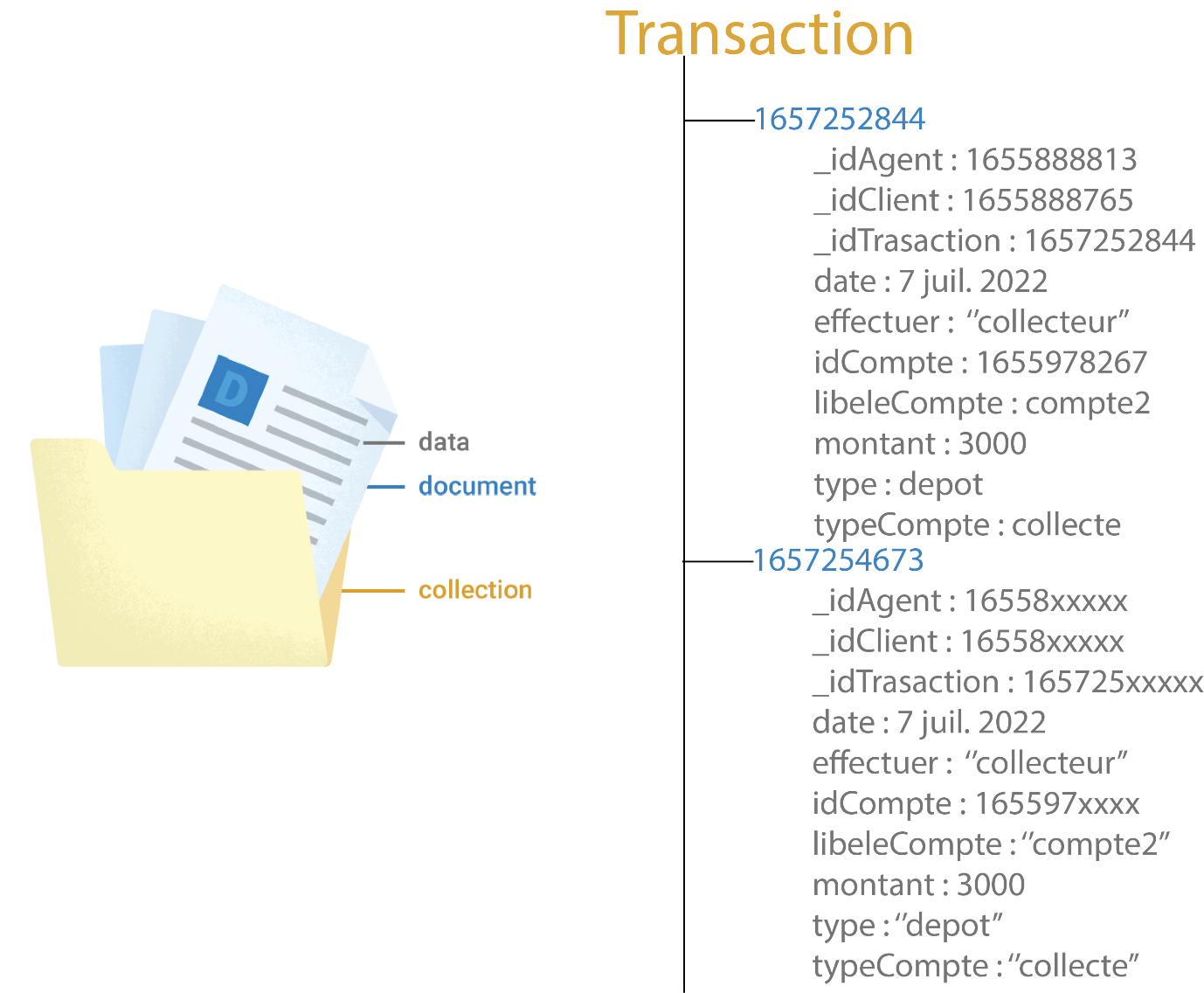


Figure 23 : représentation de la collection des Transaction

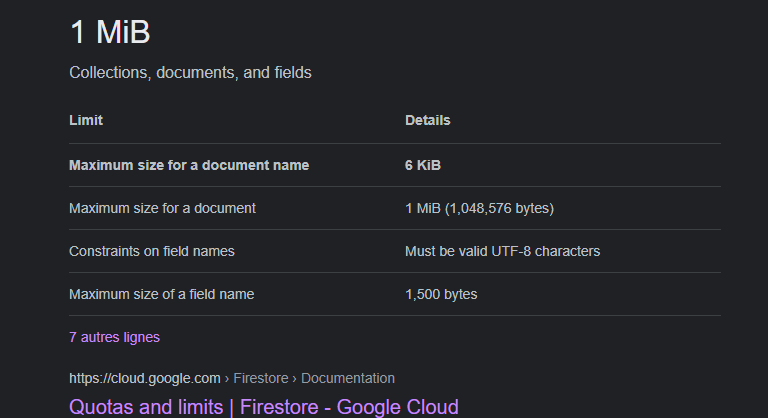


Figure 24 : règle de création d’un document

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

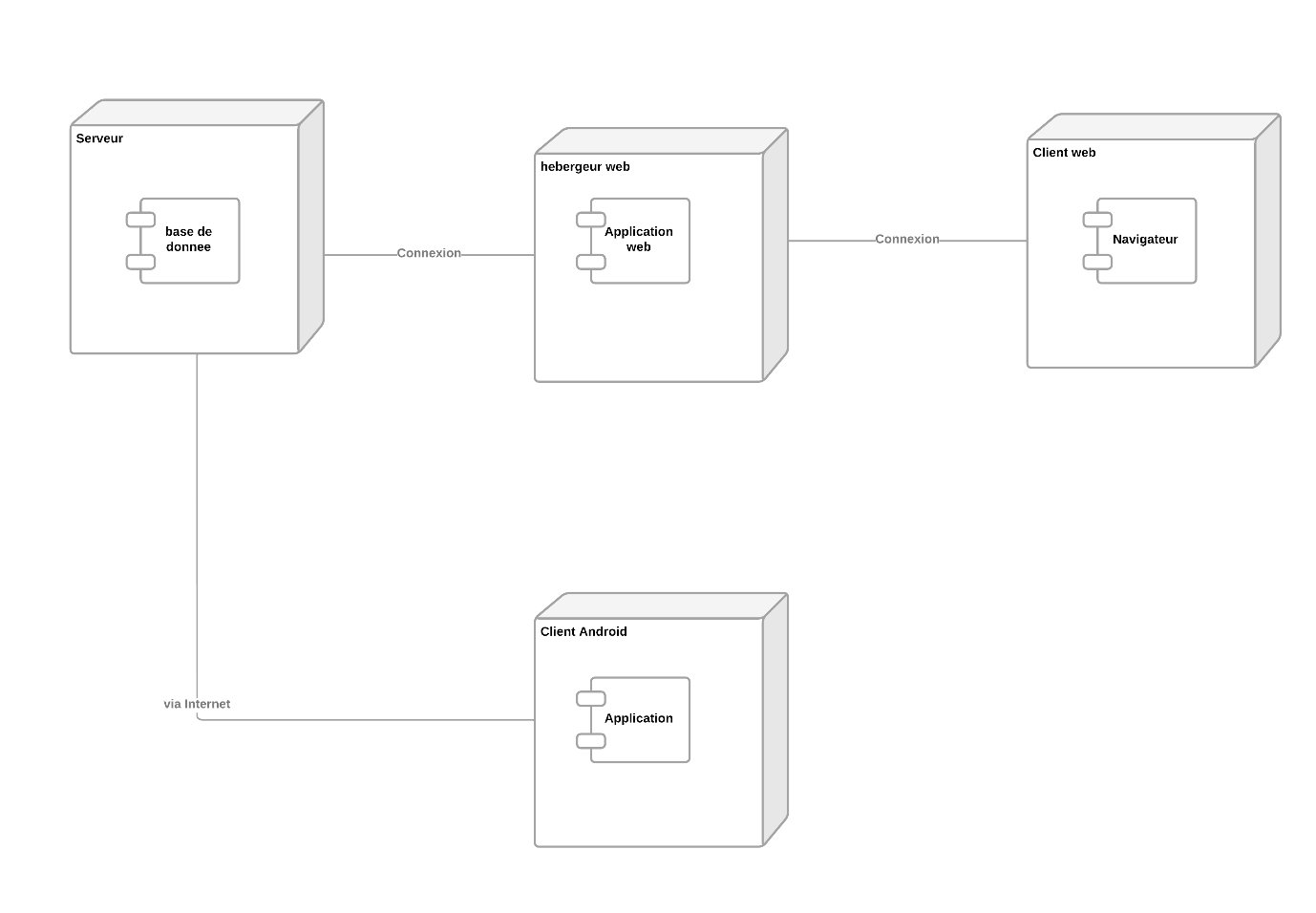


Figure 25 : Diagramme de déploiement

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

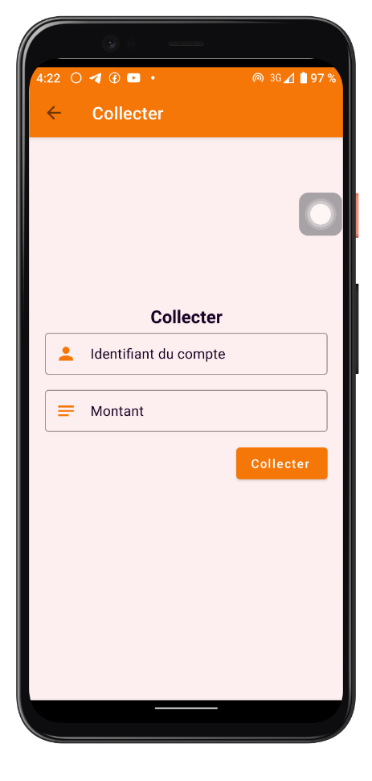
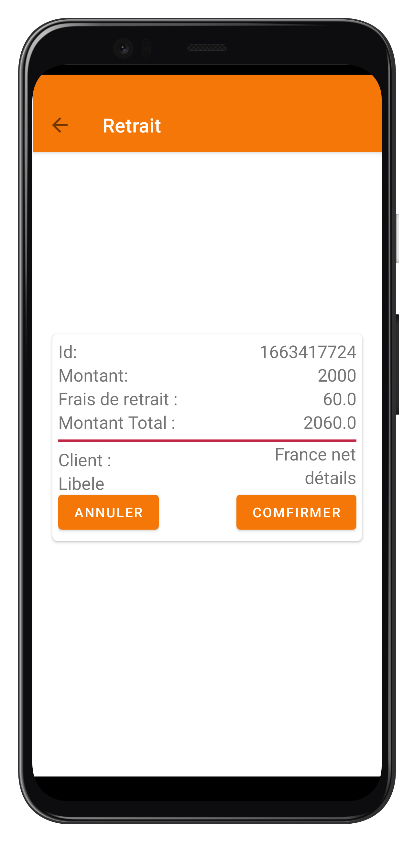
 

Figure 26 : vues : formulaires de (collecte, ouverture de compte, confirmer un retrait)

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

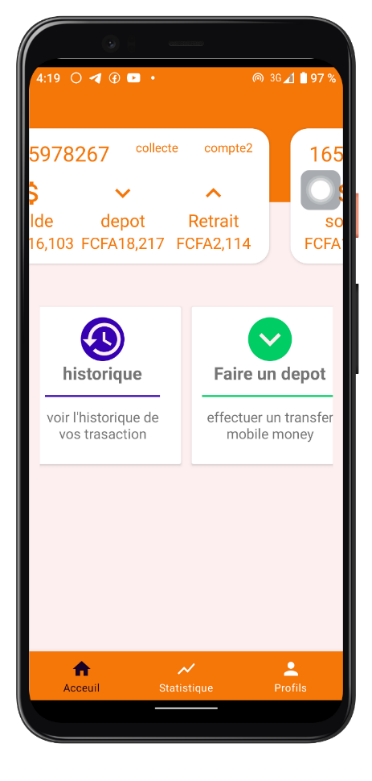


Figure 27 : vues : Les différent tableaux de bord : collecteur, client, administrateur

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

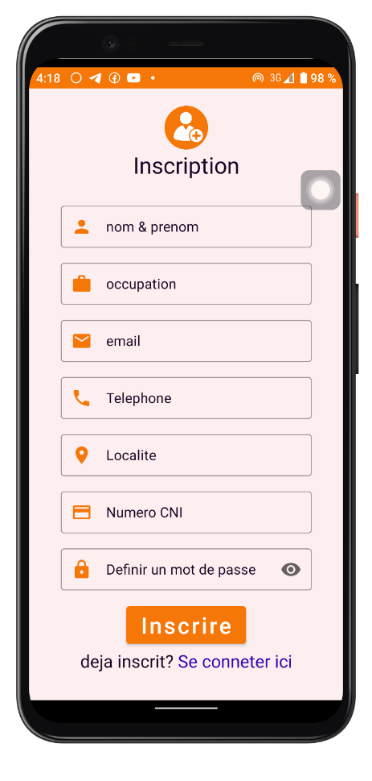


Figure 28 : vues : formulaires d’inscription et de connexion

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

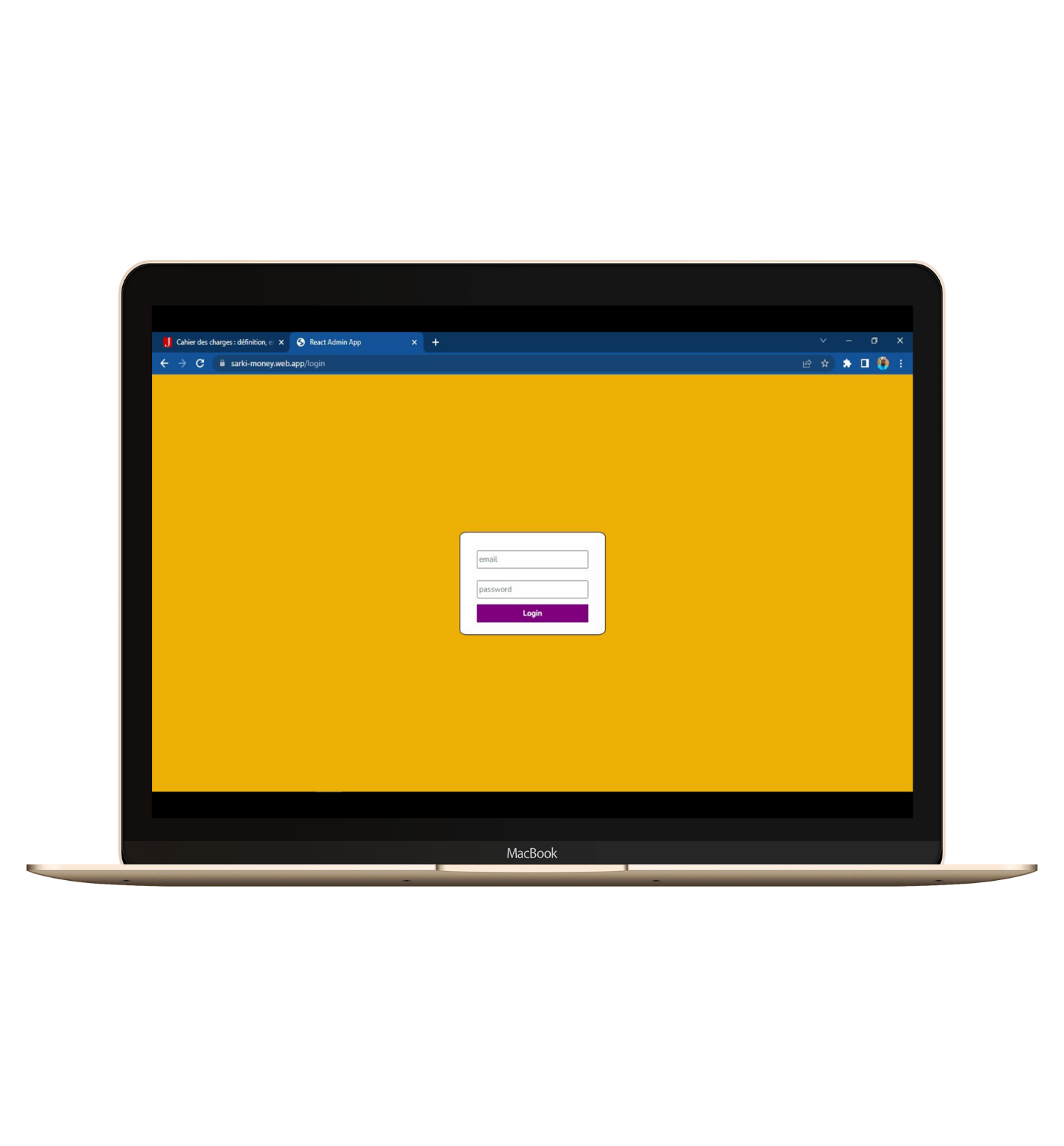


Figure 29 : Vue : formulaires de Connexion réserver à l’administrateur

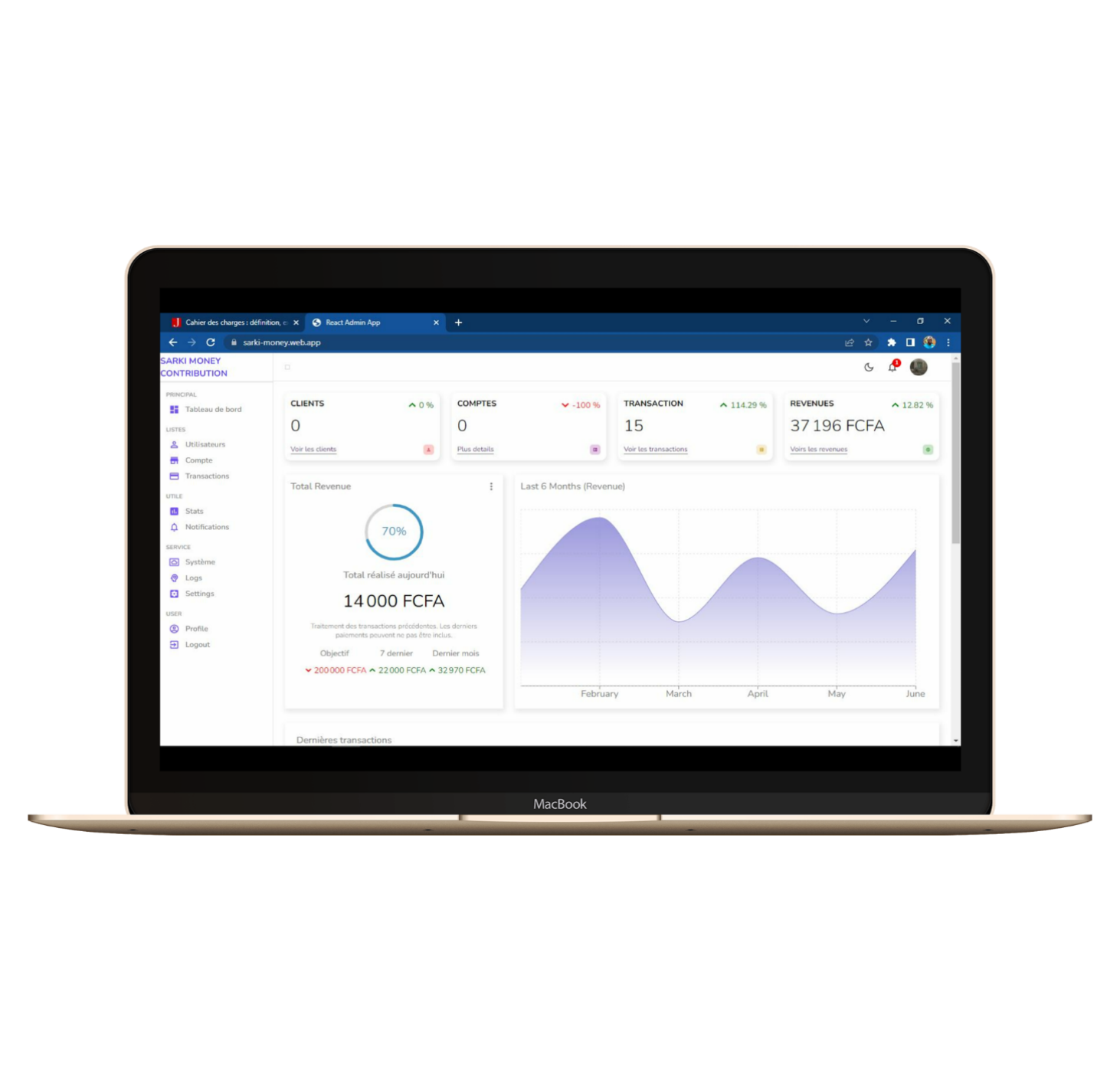


Figure 30 : vues : tableau de bord de l’administrateur sur web

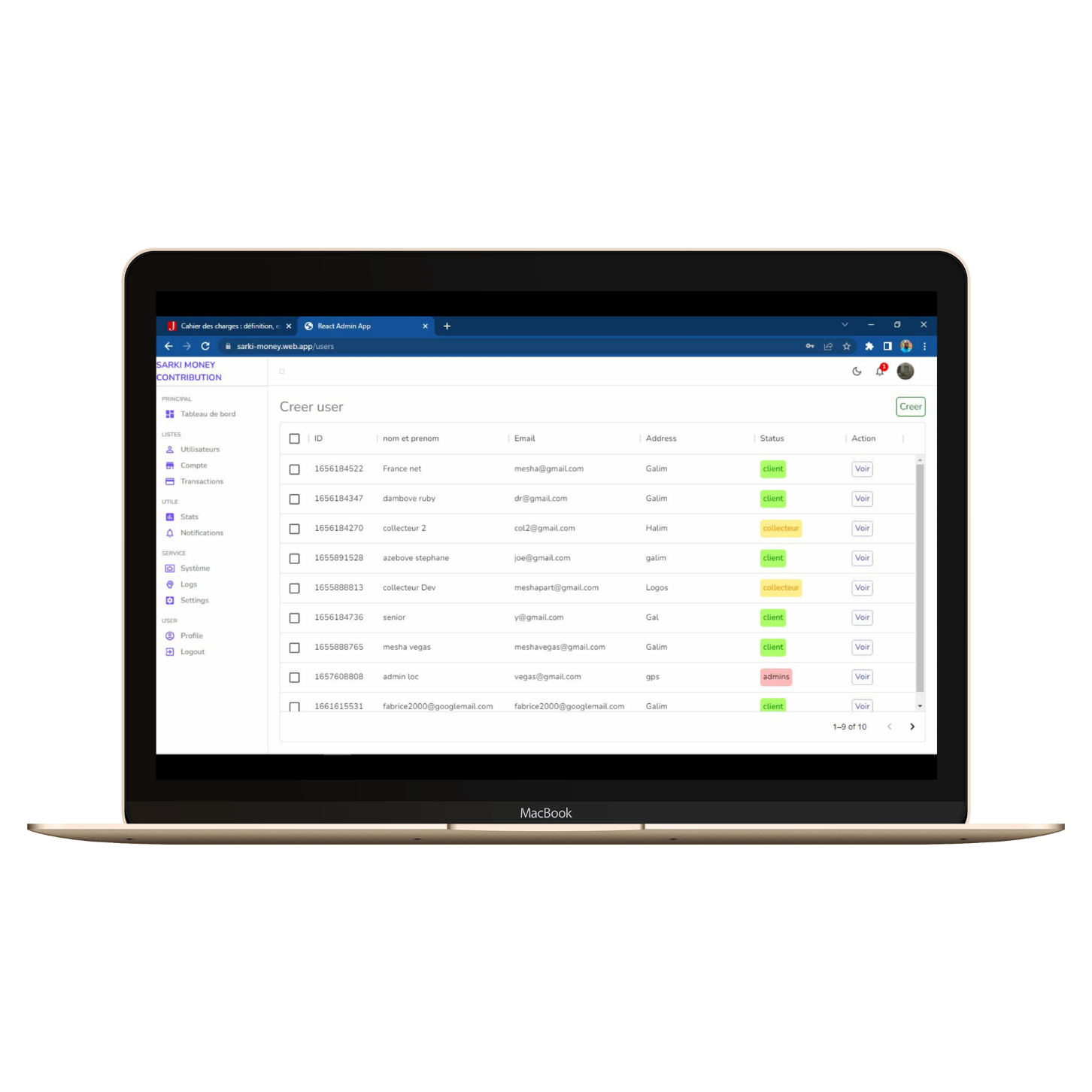


Figure 31 : vues liste des utilisateurs

#### I.2.1) Les modèles

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

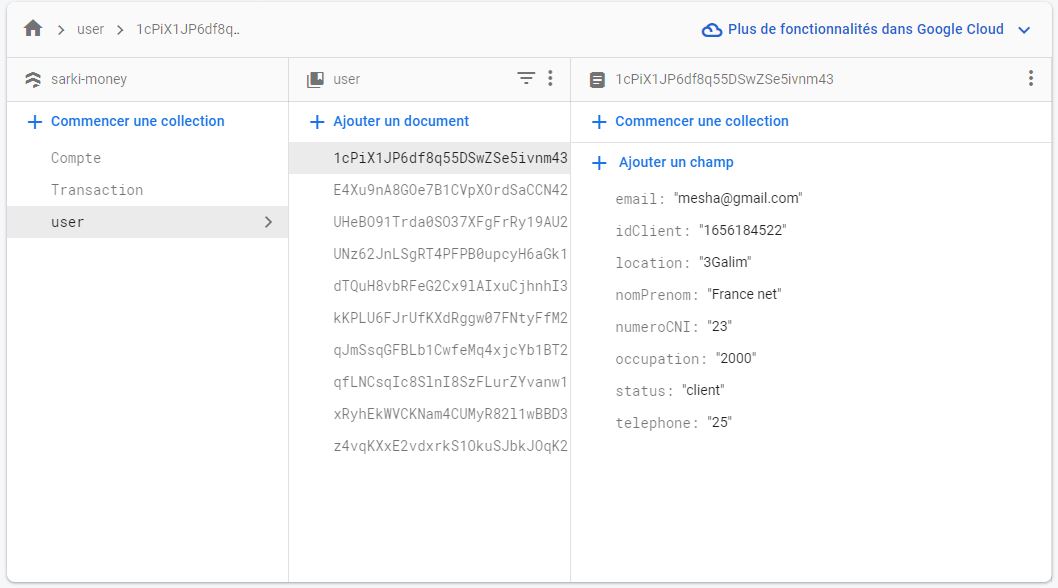


Figure 32 : collection des utilisateurs

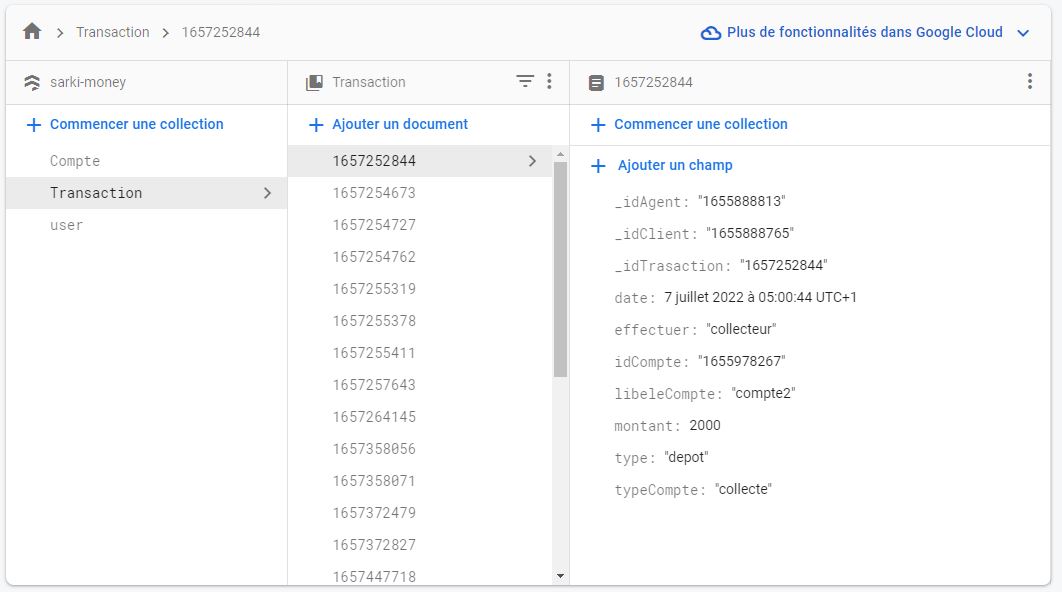


Figure 33 : collection des transactions

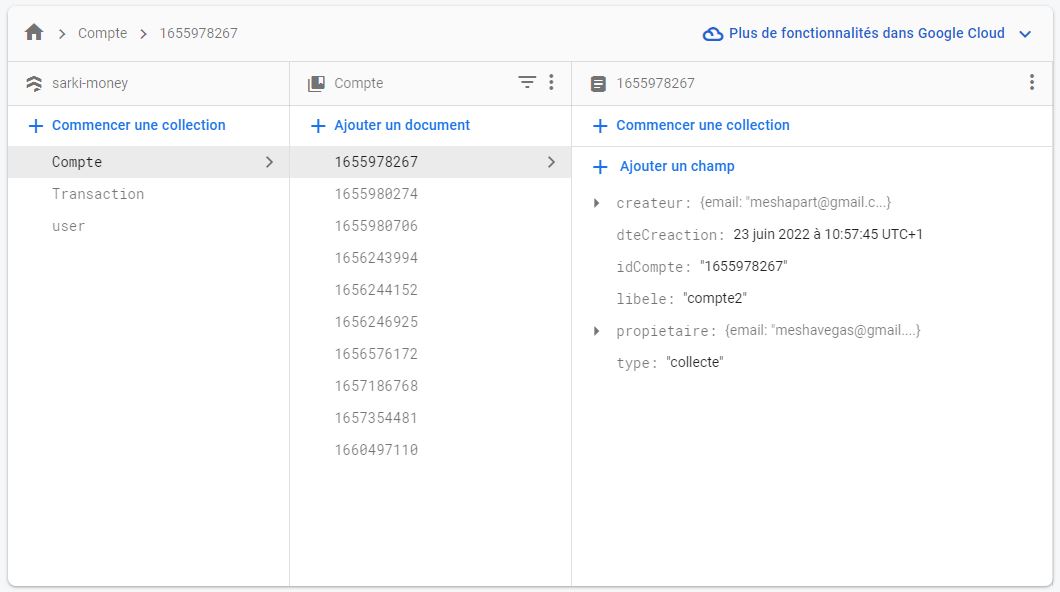


Figure 32 : collection des Comptes

# 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 49

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