**CONCEPTION D’UN OUTIL D’AIDE A LA REMÉDIATION KYC :**

**CAHIER DE CHARGES**

**CONTROLE**

**M. AYISSI AYISSI Ulrich Antoine**

Organisateur Fonctionnel junior à BC-PME SA

**COORDINATION**

**M. NDJOMO EKO Rodrigue Armel**

Chef Service de l’Organisation à BC-PME SA

Banque Camerounaise des Petites et Moyennes Entreprises



**SOMMAIRE**

[**LISTE DES FIGURES** 3](#_Toc71699101)

[**LISTE DES TABLEAUX** 4](#_Toc71699102)

[**INTRODUCTION** 5](#_Toc71699103)

[**I.** **PRÉSENTATION DU PROJET** 5](#_Toc71699104)

[**II.** **DESCRIPTION DU CONTEXTE** 5](#_Toc71699105)

[**III.** **BESOINS METIERS** 6](#_Toc71699106)

[**1.** **Besoins fonctionnels** 6](#_Toc71699107)

[a. Les besoins mobiliers 6](#_Toc71699108)

[b. Les besoins d’exploitation 6](#_Toc71699109)

[c. Fonctionnalités de l’outil 6](#_Toc71699110)

[**2.** **Besoins non-fonctionnels** 7](#_Toc71699111)

[**IV.** **BESOINS TECHNIQUES** 8](#_Toc71699112)

[**1.** **Besoins en ressources matériels** 8](#_Toc71699113)

[**2.** **Besoins en ressources logiciels** 9](#_Toc71699114)

[**V.** **OBJECTIFS VISÉES** 9](#_Toc71699115)

[**VI.** **SOLUTIONS PROPOSÉES** 9](#_Toc71699116)

[1. **Description** 10](#_Toc71699117)

[**2.** **Résultats attendus** 10](#_Toc71699118)

[**3.** **Biens livrables** 10](#_Toc71699119)

[**VII.** **DÉMARCHE D’INTERVENTION** 11](#_Toc71699120)

[**1.** **ARCHITECHTURE** 11](#_Toc71699121)

[a. Architecture 1-tiers 11](#_Toc71699122)

[b. Architecture 2-tiers 11](#_Toc71699123)

[c. Architecture 3-tiers 12](#_Toc71699124)

[**2.** **MÉTHODE** 12](#_Toc71699125)

[a. UML 13](#_Toc71699126)

[b. 2TUP (2 tracks unified process) 13](#_Toc71699127)

[c. Les phases du processus 2TUP 14](#_Toc71699128)

[**3.** **PLANIFICATION** 15](#_Toc71699129)

[**VIII.** **PERFORMANCE DU SYSTÈME** 16](#_Toc71699130)

[**1.** **La fiabilité** 16](#_Toc71699131)

[**2.** **Facilité d’utilisation** 16](#_Toc71699132)

[**3.** **Le rendement et L’efficacité** 16](#_Toc71699133)

[**4.** **La maintenabilité** 16](#_Toc71699134)

[**5.** **La portabilité** 17](#_Toc71699135)

[**IX.** **PRÉSENTATION DE L’ÉQUIPE** 17](#_Toc71699136)

[**CONCLUSION** 17](#_Toc71699137)

# **LISTE DES FIGURES**

[Figure 1:processus 2TUP 14](#_Toc71699030)

# **LISTE DES TABLEAUX**

[Tableau 1: Planification du projet 13](#_Toc70502735)

# **INTRODUCTION**

L'informatique, science de traitement automatique de l'information, constitue un domaine pratiquement incontournable dans la résolution de multiples problèmes, principalement ceux liés à la gestion optimale des organisations. Dans cette vision, nous inscrivons notre présent travail pour n'aborder que le problème lié à la conception d'un outil d’aide à la remédiation KYC (**Know Your Customer**) afin de résoudre certaines difficultés que rencontrent la BC-PME SA dans ce processus.

Le **KYC** est typiquement une politique mise en œuvre pour se conformer à un programme d'identification client selon la loi sur le Secret bancaire (consiste en l'obligation de confidentialité à laquelle sont tenus les établissements de crédit par rapport aux actes, faits et informations concernant leurs clients, et dont ils ont connaissance dans l'exercice de leur profession). Les politiques KYC deviennent de plus en plus importantes à l'échelle mondiale pour empêcher la fraude, le vol d'identité, le blanchiment d'argent et le financement du terrorisme. De ce fait, l’optimisation sur la vérification de la conformité des informations clientèles au sein de la BC-PME SA devient alors nécessaire.

# **PRÉSENTATION DU PROJET**

Dans l’optique de dynamiser nos compétences pendant notre stage, un projet nous a été assigné, sous le thème : «conception d'un outil d’aide à la remédiation KYC : cas de la **BC-PME** **SA**». En effet, par remédiation KYC nous y voyons une vérification et une correction des cas d’anomalies de données clientèles. Ces travaux incluent une gestion particulière en matière de structuration et d’organisation des dossiers clients, qui subiront un traitement logique et cohérent. Tout ceci afin de parvenir à un pourcentage de conformité meilleure. Notons aussi que ce projet s’inscrit dans le but de mettre fin aux méthodes manuelles et imparfait dans le processus de remédiation KYC à la **BC-PME SA**.

# **DESCRIPTION DU CONTEXTE**

Lors de notre passage d’imprégnation au Service de l’Organisation nous avons remarqué plusieurs anomalies présentes dans les informations clientèles. Dans le SI, de cette problématique, découle le besoin de concevoir un outil qui permettra d’assurer la mise à jour des données et informations clientèles. ». La raison qui nous oblige à mettre en œuvre ce système est que, le KYC occupe une place très importante non seulement sur le volet la Gestion de la Relation Client mais également sur le volet de la conformité précisément pour empêcher la fraude, le vol d'identité, le blanchiment d'argent et le financement du terrorisme.

# **BESOINS METIERS**

# **Besoins fonctionnels**

## Les besoins mobiliers

Nous aurons besoin de :

* **Bureau** : pour des séances de travail et des réunions d’avancement avec le chef de projet ;
* **Ordinateur portable** : celui-ci nous permettra de modéliser, programmer, compiler et de tester notre logiciel ; (Capacité minimale 8Go de RAM, DD 500Go, SE Windows 8/10) ;
* **Logiciels** : IDE, SGBD, Serveurs
* **Bloc-notes**: pour la prise des notes.

## Les besoins d’exploitation

L’application qui sera livré à la fin de ce projet, a pour but d’optimiser la remédiation KYC au sein de la BC-PME SA. Etant donné que la cellule de remédiation KYC n’a pas encore été créée, ce logiciel sera dans un premier temps octroyé au Service de l’Organisation.

## Fonctionnalités de l’outil

L’outil devra permettre à l’agent de vérification :

* De se connecter/se déconnecter sur l’outil ;
* D’effectuer une recherche d’anomalies sur les informations clientèles selon un ou plusieurs critères (champ vide, information incorrecte, information obsolète)
* D’effectuer une recherche d’anomalies sur les informations clientèles selon les contrôles sur le nom du père ou le nom de la mère (nom du père ou le nom de la mère= **INCONNU/INCONU**)
* D’effectuer une recherche d’anomalies sur les informations clientèles dont le numéro de patente est mal renseigné (nombre de caractères en moins ou exagérés) ;
* D’effectuer une recherche des anomalies sur les informations clientèles dont le numéro d’identité sociale est mal renseigné ;
* D’effectuer une recherche des anomalies sur les informations clientèles dont le numéro d’identité fiscale est mal renseigné ;
* D’effectuer une recherche des anomalies sur les informations clientèles dont le numéro du registre de commerce est expiré ou mal renseigné ;
* D’effectuer une recherche des anomalies sur les informations clientèles dont le numéro de la CNI est expiré ou mal rempli ;
* De mettre à jour les contacts des clients (téléphone, emails);
* De faire des alertes, notifications et propositions sur les anomalies
* D’effectuer des corrections (automatiques et semi-automatiques) sur les champs en anomalies et sauvegarder ;
* Faire un reporting ;
* Lancer une comparaison.

L’outil doit également permettre à l’administrateur :

* Se connecter/se déconnecter sur l’outil ;
* Créer des comptes utilisateurs ;
* Modifier des comptes utilisateurs ;
* Supprimer des comptes utilisateurs ;
* Consulter l’historique des modifications effectuées.

L’application sera utilisée par une multitude d’agent de vérification, ces agents ne sont pas susceptibles de consulter l’application en dehors des locaux de la BC-PME SA.

# **Besoins non-fonctionnels**

Il s'agit des besoins qui caractérisent le Système d’Information. Ce sont des besoins en matière de performance, de type de matériel ou de type de conception. Ces besoins concernent les contraintes d'implémentation de l’application à réaliser basées sur :

* Le langage de programmation comme JAVA EE dans notre cas.
* Le type de SGBD : MySQL pour la gestion des bases de données.
* Le système d'exploitation : Windows, Linux, MacOs.

Nous rajoutons que, l'application devra être extensible, c'est-à-dire qu'il y’ aura la possibilité d'étendre ses fonctionnalités. Il faudra aussi noter que celle-ci devra être sécurisée à la mesure où les certaines fonctionnalités telles que la création, la modification et la suppression de compte ne devront pas être accessibles à tous les utilisateurs.

Parmi les technologies nécessaires nous avons entre autre :

* **SQL** (**S**tructured **Q**uery **L**anguage): utilisé pour communiquer avec une base de données. Selon **ANSI** (**A**merican **N**ational **S**tandards **I**nstitute), c'est le langage standard pour les systèmes de gestion de bases de données relationnelles.
* **JavaScript :** est un langage de programmation léger et interprété. Il est conçu pour créer des applications centrées sur le réseau. Il est complémentaire et intégré à Java. JavaScript est très facile à mettre en œuvre car il est intégré au HTML. Il est ouvert et multiplateforme.
* **HTML5** (**H**yper**T**ext **M**arkup **L**anguage) : c’est le [langage de balisage](https://fr.wikipedia.org/wiki/Langage_de_balisage) conçu pour représenter les [pages web](https://fr.wikipedia.org/wiki/Page_web). Ce langage d’écrire de l’[hypertexte](https://fr.wikipedia.org/wiki/Hypertexte), de structurer [sémantiquement](https://fr.wikipedia.org/wiki/S%C3%A9mantique) la page, de mettre en forme le contenu, de créer des formulaires de saisie.
* **CSS** (**C**ascading **S**tyle **S**heets) : Il peut être utilisé pour concevoir une maquette. Par exemple transformer [un texte affiché sur une colonne](https://developer.mozilla.org/fr/docs/Web/CSS/Layout_cookbook/Column_layouts) en une composition avec un cadre principal et une barre latérale pour les informations reliées. Avec CSS, on peut aussi produire des animations en vue de rendre notre outil dynamique.

# **BESOINS TECHNIQUES**

Il s’agit de présenter les spécifications techniques, à savoir les besoins matérielles et logicielles qui seront nécessaire pour réaliser notre application.

# **Besoins en ressources matériels**

Les ressources matérielles minimal qui seront supporté par notre logiciel sont :

* Un ordinateur pentium 4 avec 4 giga de RAM ;
* Un serveur de Fréquence du processeur : 2.3 GHz, avec une architecture de processeurs de 64 bits, une mémoire cache de 30Mo, un disque dur de 20 terra giga et une RAM de 384 giga.

# **Besoins en ressources logiciels**

Nous aurons comme ressources logicielles :

* **Le logiciel Visual Studio Code** : un éditeur de code source gratuit créé par Microsoft pour **Windows**, **Linux** et **macOS**. Ses fonctionnalités incluent la prise en charge du débogage, la coloration syntaxique, la complétion de code intelligente, des extraits de code, la refactorisation du code et de Git intégrer.
* **L’application Enterprise Architect**: c’est un outil graphique multi-utilisateurs conçu pour aider les équipes à construire des systèmes robustes et maintenables, et en utilisant des rapports et une documentation intégrée de haute qualité. Avec des capacités de gestion des exigences incorporées, Enterprise Architect vous aide à tracer les spécifications de haut niveau à l'analyse, le design, la mise en œuvre, le test et des modèles de maintenance en utilisant UML, SysML, BPMN et d'autres normes ouvertes.
* **Oracle data base 10.g :** Il s'agit d'une base de données couramment utilisée pour exécuter des charges de travail de base de données de traitement des transactions en ligne (OLTP), d'entreposage de données (DW) et mixtes (OLTP et DW). Oracle Database est disponible par plusieurs fournisseurs de services sur site, sur le cloud ou en tant qu'installation de cloud hybride. Il peut être exécuté sur des serveurs tiers ainsi que sur du matériel Oracle.
* **Windows :** l’outil tournera uniquement sur ce système d’exploitation.
* **Les navigateurs**: qui devront être supportés lors du déploiement de l’application sont : Opéra, Mozilla Firefox et Google Chrome.

# **OBJECTIFS VISÉES**

Ce projet vise à :

* Appliquer les concepts de la Modélisation UML;
* Appliquer les notions de JavaScript et PrimeFaces ;
* Implémenter une interface ;
* Faciliter et automatiser la remédiation KYC;
* Contrôler la mise à jour d’informations clientèles.

# **SOLUTIONS PROPOSÉES**

La solution que nous proposons, est celle-là qui vient répondre au problème de vérification des informations clientèles, qui passe par une compréhension proprement dite de celui-ci, ensuite par sa compréhension, et enfin aux spécifications des attentes face à sa résolution.

# **Description**

Il s’agira avant tout dans notre cas d’une application « **DESKTOP** », présentant une interface d’administration à deux dimensions, en fonction des privilèges de postes, notamment: un administrateur central pouvant être le Chef service de l’Organisation et Développement et des agents de vérifications. En gros, l’application sera à mesure d’assurer les fonctions suivantes:

* Gestion des informations clientèles;
* Gestion des statistiques;
* Gestion des utilisateurs.

L’application sera Développé à partir de **JavaScript** avec PrimeFaces implémenté à travers une interface graphique et les différents modules qui seront développés à cet effet sont :

* Module d’Administration ;
* Module d’Extraction et Analyse ;
* Module de Reporting ;
* Module de Correction ;
* Module Alertes et Emails.

# **Résultats attendus**

Au terme de ce projet, nous devrions avoir mis sur pied une application web faite à travers de la technologie Angular, et garant de son impacte aussi bien en milieu éducatif qu’en milieu professionnel.

**Sur le plan personnel**: L’application devra améliorer le processus de remédiation KYC, et le souci d'informatiser un système manuel en apportant un savoir-faire pour alléger les tâches liées à la gestion des informations clientèles.

**Sur le plan sociotechnique**: Actuellement l'évolution de la technologie nous oblige à intégrer l'informatique dans le mécanisme de création des richesses en entreprise, au moyen des ordinateurs, raison pour laquelle nous devons mieux appréhender les principes et techniques de travail, afin de garantir une meilleure transcription des problèmes rencontres en environnement réel, à celui du numérique.

# **Biens livrables**

Ce sont ici des « outputs » qui définiront de manière concrète l’achèvement du projet. Ce sont :

* **Le cahier des charges** : C’est un document contractuel. Il résume le périmètre fonctionnel et technique du projet attendu par le client sur lequel les développeurs s’engagent de le réaliser ; Il décrit précisément les besoins auxquels le prestataire ou le soumissionnaire doit répondre, et organise la relation entre les différents acteurs tout au long du projet.
* **Le cahier d’analyse :** C’est un document qui illustre au travers des différents diagrammes tels que le diagramme de cas d’utilisation, le diagramme de classe, le diagramme séquence, et le diagramme d’activité, le dynamisme du système vis-à-vis des acteurs.
* **Le cahier de conception :** ce document rassemble les critères et les orientations ayant servi à la conception du site. Il fournit le cadre conceptuel indispensable à toute modification de l'architecture.
* **L’application proprement dite** : Qui est l’application informatique à réaliser, et qui devra respecter les contraintes définies dans le cahier des charges et le cahier d’analyse.
* **La documentation**: C’est la description textuelle qui accompagne l’application réalisée. Elle explique comment l’application fonctionne, et/ou comment on doit l'employer ou la déployer.
* **Cahier de tests :** un document fondateur de la relation contractuelle entre les équipes et le chef de projet ou prestataires et clients. Il doit donc être réalisé en amont de l'action souhaitée (phase de tests, lancement de projet, etc.…), tout au long de celle-ci, et même en post-projet.
* **Cahier de recette :** C’est un document de jalons très important qui contient le rapport des tests dont le but est de s’assurer de la fiabilité et de la robustesse du logiciel à fournir.

# **DÉMARCHE D’INTERVENTION**

## **ARCHITECHTURE**

### Architecture 1-tiers

Cette architecture Tiers est employée par des applications autonomes pour stocker des données, et pour montrer les résultats. Les applications autonomes fonctionnent sur un ordinateur simple, fournissent une interface utilisateur, manipulent toute entrée, valident des données, et maintiennent la base de données. Une application qui utilise l'architecture 1-Tiers est limitée par la connectivité et interopérabilité limitées.

### Architecture 2-tiers

### 

Encore appelée l’architecture « **client-serveur** » est assez simple. Vous avez d’un côté le client et de l’autre le serveur. Ce genre d’architecture peut se faire sur tout type d’architectures matérielles interconnectées. Elle fonctionne de la manière suivante : le client demande un service au serveur, le serveur reçoit cette requête, il effectue un traitement, et renvoie la ressource demandée par le client.

Les avantages d’une architecture client-serveur c’est que tout d’abord les ressources sont centralisées sur le serveur. Il est donc plus simple de gérer les ressources communes aux utilisateurs comme la base de données par exemple.

De plus, cette architecture est plus sécurisée étant donné que le client dispose de moins de point d’entrée pour accéder aux données. L’architecture 2-tiers dispose d’un grand inconvénient, vu qu’il est le cœur du réseau, si celui-ci tombe en panne, tout le système fait pareil.

### Architecture 3-tiers

Dans l’architecture 3-tiers, un nouveau niveau fait son apparition. En effet, nous avons toujours le niveau 1 qui est le client. Le client est très léger étant donné qu’il n’a aucun rôle de traitement. Au niveau 2 nous avons le serveur d’application et enfin, au dernier niveau le serveur de base de données.

Les points importants d’une application 3-tiers:

* Le client ne sert qu’à requêter et à afficher les réponses du serveur.
* Le serveur lui s’occupe des calculs et même de requêter des serveurs  
  additionnels.

Les avantages d’une architecture 3-tiers sont nombreux. Tout d’abord cette  
architecture étant plus divisé permet d’avoir du point de vue du développement, une spécialisation des développeurs selon le niveau de l’application. Cette architecture offre une flexibilité beaucoup plus importante que l’architecture 2-tiers. En effet, la portabilité du tiers serveur permet d'envisager une allocation et/ou modification dynamique aux grés des besoins évolutifs au sein d'une entreprise.

Compte tenu de ces deux architectures, nous avons opté l’architecture 3-tiers car elle permet de diviser les tâches et par conséquent d’avoir des développeurs spécialisés sur un des trois niveaux d’abstraction. De plus, la flexibilité qu’offre ce genre d’infrastructure est à prendre en considération surtout si vous travaillez sur un projet qui peut être amenée à évoluer.

# **MÉTHODE**

« La méthode est l'ensemble des opérations intellectuelles par lesquelles une discipline cherche à atteindre les vérités qu'elle poursuit, les démontre et vérifie »

Dans le présent travail, nous aurons à utiliser la démarche 2TUP, un processus de  
développement construit sur UML, qui est un Langage de Modélisation Unifiée, destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer le point de vue en passant par trois niveaux d'abstraction qui sont :

* Niveau Conceptuel ;
* Niveau Logique ;
* Niveau Physique ;

### UML

**UML** (**U**nified **M**odeling **L**anguage) ou langage de modélisation unifié, se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. **UML** permet de modéliser l'ensemble des données et des traitements en élaborant des différents diagrammes. En clair, il ne faut pas designer **UML** en tant que méthode **(Il manque la démarche)** mais plutôt comme une boite d'outils qui sert à améliorer les méthodes de travail.

**UML** dans sa version 2 s'articule autour de treize diagrammes, chacun d'entre eux est dédié à la représentation d'un système logiciel suivant un point de vue particulier. Ces diagrammes sont regroupés dans deux grands ensembles :

* Les diagrammes structurels
* Les diagrammes de comportement.

Seulement neuf de ces diagrammes retiendront notre attention tout au long de ce travail notamment :

* Le diagramme de classe;
* Le diagramme de séquence ;
* Le diagramme de cas d'utilisation ;
* Le diagramme d'activités ;
* Le diagramme de composants ;
* Le diagramme d’objet ;
* Le diagramme d’état-transition ;
* Le diagramme de package ;
* Le diagramme de déploiement.

### 2TUP (2 tracks unified process)

Un processus définit une séquence d'étapes ordonnées qui permet de produire un système logiciel ou de faire évoluer un système existant. Un processus est décomposé suivant l'axe de développement technique et l'axe de gestion du développement.

2TUP est un processus unifié qui a pour but d'apporter une réponse aux contraintes de changement fonctionnelles et techniques qui s'imposent aux systèmes d'information. 2TUP 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 le système. On obtient un processus de développement en Y comme l'illustre la figure ci-dessous. Il faut noter que le processus 2TUP commence par une étude préliminaire. Dans cette étude préliminaire, il s'agit d'identifier les acteurs qui vont interagir avec le système, les messages qu'échangent les acteurs et le système, puis à produire un cahier de charges et enfin à modéliser le contexte.



**Figure 1:processus 2TUP**

### Les phases du processus 2TUP

Le processus s’articule autour de trois phases essentielles :

* **Branche fonctionnelle ou « gauche »**

Elle vise la capture des besoins fonctionnels et l'analyse des spécifications fonctionnelles de manière à déterminer ce que va réaliser le système en termes de métier. C'est ici, qu'on identifie et dégage toutes les fonctionnalités du système à réaliser.

* **Branche technique ou « droite »**

Elle permet la capture des besoins non fonctionnels. Il s'agit essentiellement des contraintes que l'application doit prendre en compte comme par exemple les contraintes d'intégration, les contraintes de développement et les contraintes de performances.

* **Phase de réalisation**

Cette phase est la fusion des deux précédentes et mène à la conception applicative et à la solution adaptée aux besoins des utilisateurs. Elle concerne les étapes de la conception préliminaire, la conception détaillée, le codage et les tests puis l'étape de recette.

# **PLANIFICATION**

Elle définit la livraison de chaque livrable sur une période bien définie :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BC-PME SA : DEVELOPPEMENT D’UNE APPLICATION D’AIDE A LA REMEDIATION KYC | | | | | |
| NOM DE LA TACHE | | | Durée (J) | Date  Début | Livrables |
| **LANCEMENT DU PROJET** | | | **1** | **01/04/2021** |  |
| **Phase 1** | **ETUDE DE FAISABILITE** | **20** | | **02/04/2021** |  |
|  | Analyse de l’existant | 10 | |  | Rapport d’analyse |
| Analyse des besoins | 10 | |
| **Phase 2** | **ANALYSE ET CONCEPTION** | **45** | | **22/04/2021** |  |
|  | Rédaction du Cahier de charges | 7 | |  | Cahier de charges |
| Analyse technique et fonctionnelle | 15 | | Cahier d’analyse |
| Conception | 23 | | Cahier de conception |
| **Phase 3** | **IMPLEMENTATION ET DEVELOPPEMENT** | **60** | | **06/06/2021** |  |
|  | Développement du module d’Administration | 9 | |  | Code source, Documentation Explication du code  Base de données. |
| Développement du module d’Extraction et Analyse | 9 | |
| Développement du module de Reporting | 8 | |
| Développement du module de Correction | 8 | |
| Développement du module Alertes et Emails | 8 | |
|  | Front End | 18 | |  |
| **Phase 4** | **INTEGRATION ET TESTS** | **30** | | **05/08/2021** |  |
|  | Tests unitaires | 4 | |  | Cahier de tests |
| Tests fonctionnels | 4 | |
| Tests d’intégration | 4 | |
| Correction des bugs | 18 | |
| **Phase 5** | **LIVRAISON DE L’APPLICATION** | **21** | | **04/09/2021** |  |
|  | Installation de l’application | 10 | |  | Cahier de recette  Guide d’utilisation |
| Réunion de livraison de l’application | 11 | |

**Tableau 1: Planification du projet**

**NB :** Nous comptabilisons donc une durée totale de 05 mois, pour la finalisation du projet.

# **PERFORMANCE DU SYSTÈME**

Nous avons trois axes d’études dans le cadre des performances :

## **La fiabilité**

Notre outil devra tenir compte des critères suivants :

* Maturité
* Tolérance aux fautes
* Système de sauvegarde (Le système doit être restauré à une date ultérieure en cas de problème).

## **Facilité d’utilisation**

Il faudra gérer :

* La facilité de compréhension ;
* La facilité d’apprentissage ;
* La facilité d’exploitation/d’utilisation ;
* Bonne ergonomie

L’ensemble des pages de notre outil doit être accessible en 3 clics maximum depuis la page d’accueil. Le logiciel doit donner une image moderne de l’entreprise.

## **Le rendement et L’efficacité**

Le rendement d’utilisation étant un élément important, il doit également être pris en compte à travers les aspects suivants :

* Comportement vis-à-vis du temps ;
* Comportement vis-à-vis des ressources ;

Le serveur d’applications doit durer 5 ans avant le remplacement et le temps de chargement des pages doit être au plus de deux secondes.

## **La maintenabilité**

Les caractéristiques utilisées ici sont les suivants :

* Facilité d’analyse ;
* Facilité de modification ;
* Stabilité ;
* Testabilité ;

Le logiciel doit recenser les erreurs (Facilité d’analyse – solution fichier log). Le logiciel doit pouvoir démarrer même en cas de problème.

## **La portabilité**

Les caractéristiques étudiées :

* Facilité d’adaptation ;
* Facilité à l’installation ;
* Interchangeabilité ;

Le script d’installation doit s’installer en moins de 5 minutes. Les fichiers produits générés doivent être compatibles avec Word pour Mac et Windows

# **PRÉSENTATION DE L’ÉQUIPE**

Dans le cadre de ce projet, l’équipe est constituée de 5 membres notamment :

* M. NDJOMO EKO Rodrigue: Coordinateur ;
* M. MFOKOUE LETUTOUR Brice : Superviseur ;
* M. AYISSI AYISSI Ulrich : Contrôleur;
* WAMBA FOKOU Gabin Loïk: AMO (Assistant DE MAITRISE D’œuvre)
* MADA NOUPELE Vanelle Synthia: AMO (Assistant DE MAITRISE D’œuvre)

# **CONCLUSION**

Parvenu au terme de notre travail, il était question de présenter le contexte global du projet, exprimer clairement les objectifs du projet et les critères qui permettront de le considérer comme réussi, tant en quantité qu'en qualité, et de formaliser le besoin de la BC-PME SA qui est de faciliter le processus d’aide à la remédiation KYC.

Toutefois nous avons défini le contexte du projet, puis nous avons présenté les objectifs visés. Par la suite, nous avons présenté la solution proposée, puis décrit la démarche d’intervention qui sera utilisée pour pallier au problème. Enfin nous avons énuméré les performances du système, puis présenté l’équipe projet. Nous notons que l’outil à concevoir devra être fiable, facile à utiliser, il doit avoir un bon rendement. L’efficacité, la maintenabilité et la portabilité doivent également être prise en compte par l’outil à réaliser.

À l’issu de l’élaboration de ce cahier des charges, nous avons pris en compte toutes les démarches techniques auxquels nous seront confrontées tout au long de la conception et du développement de **l’application d’outil d’aide à la remédiation KYC.**