**sommaire**

[DEDICACE ii](#_Toc476724019)

[REMERCIEMENTS iii](#_Toc476724020)

[LISTE DES FIGURES iv](#_Toc476724021)

[LISTE DES TABLEAUX v](#_Toc476724022)

[RESUME vi](#_Toc476724023)

[INTRODUCTION GENERALE 7](#_Toc476724024)

[CHAPITRE I: DOSSIER D’ANALYSE 9](#_Toc476724025)

[**INTRODUCTION** 9](#_Toc476724026)

[**I.** **Contexte** 10](#_Toc476724027)

[**II.** **Problématique** 10](#_Toc476724028)

[**III.** **Solutions probables** 11](#_Toc476724029)

[**IV.** **La Cible** 11](#_Toc476724030)

[**V.** **Impacts de la solution** 11](#_Toc476724031)

[**VI.** **Cahier de charge** 12](#_Toc476724032)

[**VII.** **METHODOLOGIE** 16](#_Toc476724033)

[**VIII.** **MODELISATION DE LA SOLUTION** 18](#_Toc476724034)

# DEDICACE

AU CORPS DES ENSEIGNANTS DE L’IAI

# REMERCIEMENTS

Nous tenons à remercier tout particulièrement :

* **M. Armand Claude ABANDA,représentant Résident** de l’**IAI au Cameroun Centre d’Excellence Technologique Paul BIYA**, pour les efforts fournis pour mettre à notre disposition les meilleures conditions scolaires ;
* **M. OMGBA,** notre enseignant de POOA(Programmation Oriente Objet Avance ) pour les efforts fournis pour mettre à notre disposition les connaissance requise pour le codage de l’application dans le langage C++ avec la librairie graphique Qt, sans oublié les notions importantes de la POO et des bonnes regles de developpement qu’il a mis a notre disposition;
* Tout le Corps Administratif et Professoral de **l’IAI Cameroun** , pour les connaissances additionnelles qui ont peut servir;
* Tous mes camarades pour leur disponibiles, leurs aides intellectuelles et surtout pour leurs critiques.
* A tous ceux qui ont participés d’une manière ou d’une autre à la réussite de ce travail;

# LISTE DES FIGURES

[Figure 2: processus de développement 2TUP 17](#_Toc476718618)

[Figure 3: étude préliminaire 18](#_Toc476718619)

[Figure 4:formalisme des use case 18](#_Toc476718620)

[Figure 4:diagramme class de conception 20](#_Toc476718621)

[Figure 5: ERD :Modele Physique de donnees 21](#_Toc476718622)

# LISTE DES TABLEAUX

[Tableau 4 :Ressources humaines 13](#_Toc476718605)

[Tableau 5 :Ressources matérielles 14](#_Toc476718606)

[Tableau 6 :Ressources logicielles 14](#_Toc476718607)

[Tableau 7 :Budget total du projet 14](#_Toc476718608)

# RESUME

Les objectifs d’une entreprise de services sont d’une part la satisfaction de ses clients à travers des services répondant à leurs attentes et réalisés dans les budgets et les délais impartis, et d’autre part l’optimisation des ressources afin que l’entreprise puisse réaliser ses travaux de manière rentable par le biais de solution et methode de l’informatique. C’est dans cette optique que L’IAI (Institut Africain de l’Informatique) forme des ingenieurs à la comception et la realisation des applications et la creation des entreprises. Il nous a été demande durant notre 3e année de la formation en genie logiciel, de s’appliquer a travers un travail pratique du cour de POA (programmation orienté objet avancé ) dans le langage C++ en utilisant la library Qt.

# INTRODUCTION GENERALE

La réussite des societes de personnes, dépend dans une large mesure, de leur capacité à gérer leur système d’information. Il est important qu’au sein d’un groupe de personne qu’il y reigne de la confiance. Etant donne que nous nous situons dans le cadre de la gestion financiere, il est important de reduire les erreurs de calcul , d’apporter de la transparence et de prendre des bonnes decisions, afin que la societe puisse arriver a maturité et survivre pendant longtemps.

Il s ‘agit de déployer un système d’information autour d’une application permettant à celle d’assainir la structure financiere telsque les tontines.

Au vue des contraintes liées à l’exécution du projet, nous avons choisi de baser notre développement selon un aspect fonctionnel et une architecture technique bien définie. Le présent rapport sera structuré en une seul parties couvrant l’ensemble des aspects de notre travail :

* 1erpartie, le dossier d’analyse: nous allons presenter le contexte a nos jours, nous allons recencer les problemes majeurs et proposer la ou les solutions propables, ainsi que la présentation globale de notre solution et son analyse de bout en bout en utilisant les outils informatiques qui nous ont été enseignés a l’IAI;

CHAPITRE I : DOSSIER INSERTION

# CHAPITRE I: DOSSIER D’ANALYSE

## INTRODUCTION

Nous nous rendons à la rédaction du dossier d’analyse y compris le cahier de charge qui montre de façon précise et concise les attentes et les exigences de la structure liée à notre thème. Nous allons suivre la demarche d’analyse 2TUP et les diagrammes de UML pour pouvoir etudier l’existant, comprendre les besoins fonctionnels et operationnels des futurs utilisateur, concevoir modele modele sous forme de diagramme et enfin realiser, tester, mettre en service l’application MAMITON, qui est gestionnaire des tontines a l’africaine.

## Contexte

Enoncé :

De nos jours en afrique, il est tres coutumié de voir un groupe de personnes (des amis, des feres, des camarades et des colleges de service) se reunir a un endroit pour gerer leurs gains. Cet endroit est comme une fraternité couramment appele tontine. Contrairement a la banque ou on conserve les biens des personnes qui n’ont aucune relation entre elles,dans les tontines on y entre par le biais de quelqu’un déjà dans le système. On peut y faire partir si on est l’ami ou l’ami de l’ami, le frere du frere , … , le camarade du l’ami d’un adherants. Ces derniers sont directement liés ou par le biais du serie d’adherants aussi liés. dans les tontines la confiance est l’une des choses les plus souhaites. On est reponsable de chaque personne qu’on amene dans la tontitne

Le fonctionnement des tontines à l’africaine est simple : un groupe de personnes décident de se réunir régulièrement pour mettre leur épargne en commun. Chacun cotise une somme fixe, déterminée à l'avance. Par exemple : 10 personnes cotisent 10 000 fcfa. A chaque rencontre (séance) une seule personne empoche l'intégralité du magot (lot), soit 100 000 fcfa. Et ainsi de suite, jusqu'à ce que tous les adhérents aient «bouffé la tontine». En plus de ce rôle informel d’épargne et de crédit, les tontines jouent un important de cohésion sociale entres les membres.   
  
Ayant constaté qu’il n’existait pratiquement pas de logiciel spécialisé dans la gestion des activités d’une tontine. L’idée est de faciliter la saisie des opérations afin de permettre un meilleur suivi des dettes et des cotisations. Ceci permettant donc d’éviter l’endettement important de certains membres car il s’agit d’un risque important à gérer pour les tontines : la perte de confiance crée par les membres défaillants, sachant qu’en cas d’abus les recours en justice sont presqu’impossibles.

## Problématique

* Comment éviter la fraude, la tromperie et les detournements des fonds dans les tontines ?
* Comment eviter les endettes execif des membres ?
* Comment optimiser la prise de decisions et deminuer les erreurs humaines au sein de la tontine?

## Solutions probables

Après avoir pris connaissance de problèmes présentés dans ce contexte, nous proposons comme solution une application Win32 appelé **MAMITON** à travers laquelle les membres de la tontine pourront

* Tracer les encaissements et les decaissements d’argent (cotisations, emprunt, aide) avec une forte transparence,
* Empecher l’accumulation des emprunts
* Gerer le planing des tontines,
* Etre notifier sur les etats critiques de la tontine ou d’un membre (par exemple approche de la date de cloture d’une tontine, longue absence d’un membre)
* Avoir un etat sur les personnes ayant déjà bouffe leur cagnote dans une tontine
* Gerer la liste de membres dans chaque tontine
* Automatiser les suspensions des membres en fonction des regles disciplinaires et detecter les deserteurs en fonction des absences.
* Gerer les absences , les retards et appliquer automatiquement les sensions (les prix a payer)
* Imprimer les etats (informations a un moments precis) d’un niveau d’avancement de la tontine

## La Cible

Notre cible est constituée des petits et moyennes tontines auquels peut y adherer des jeunes, et des parents.

## Impacts de la solution

Afin de démontrer la pertinence du contexte énoncé et utilité des solutions proposés, nous avons cités quelques impacts favorables apportés.

Sur le plan marketing

* L’automatisation apporte de la valeur au label de la societe ou secte (du groupe de personne) ayant initier les differentes tontines.

Sur le plan social et politique

* Apporte la confiance chez les membres de la tontine a cause de la tracabilise des mouvement de caisse et des actes de chaque membres

Sur le plan commercial et financier

* Exactitude des calculs et des bilans financiers
* Reductions et supressions du budget dedier aux registres et

## Cahier de charge

### Objectifs General

L’application a pour objectif l’automatisation, la transparence et la tracabilité des actions pouvant etre effectuées dans un tontine.

### Exigences

Le produit livré sera jugé acceptable si toutes les exigences suivantes sont respectées :

* Contrainte de délai : deux semaines de realisation;
* Contrainte de fonctionnalité : garantir la realisation des deux tiers au moins des fonctionnalités prevues deux semaines apres le lancement du projet;
* Contrainte d’intégrités : l’application doit être sécurisée, gérer les rôles et les droit d’accès, discretion totals sur les informations personnels, respecter les règles de moralité civique et des droits de l’homme;

### Objectifs spécifique

##### Les besoins fonctionnels

Les besoins fonctionnels listent les opérations réalisables de notre application. Ce sont des besoins spécifiant un comportement d'entrée / sortie du système.

L’application se décompose en plusieurs modules et doit réaliser les opérations suivantes :

* **Le module des de contrôle d’accès**
  + Authentification (connexion à son espace personnel)
* **Module de gestion de la discipline**
  + Gerer les absences, les retards et autres mauvais comportements a l’iinterieur ou a l’exterieur de la tontine
* **Module de gestion des encaissements et des decaissements**
  + Gerer les cotisations, les emprunts, les demandes d’aides
* **Module de gestion des utilisateurs, des comptes**
  + Mise à jour du profil des membres.
  + Suspension temporaires ou definitives des membres
  + Creation des membres
* **Module de tontines**
  + Production des rapports et bilans financiers.
  + Affiliation d’un membres a un tontine
  + Creation des tontines
* **Module de gestion notification**

##### Les besoins opérationnels

**Les besoins opérationnels** représentent les besoins non fonctionnels, qui caractérisent le système comme la performance ainsi que la sécurité et l’ergonomie du système. Ces besoins peuvent être énoncés suivant des plans de classifications.

* **L'ergonomie des interfaces**

L’interface d'une application est délicate,il doit être simple, claire Facile à manipuler, compréhensible, intuitive. La manipulation de l'interface ne doit pas nécessiter des connaissances poussées , il être bien organisés du point de vue graphique, le choix des couleurs, et des styles.

* **Robustesse et Finesse**

L'application doit assurer le stockage permanent des informations des utilisateurs inscrits, ainsi qu'assurer une bonne gestion d'erreurs, doit consommes le minimum des ressources en conservant le maximum de rapidité possibile.

* **Sécurité**

L'application doit garantir à l'utilisateur connecté l'intégrité et la confidentialité de ses données.

### Déroulement du projet

Nous allons ainsi donc vous présenter tour à tour les ressources, l’organisation du projet et les livrables.

#### Estimation des ressources nécessaires

Tableau 4 :Ressources humaines

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***NOMBRE*** | ***FONCTION*** | ***PERIODE (jrs)*** | ***SALAIRE JOURNALIER*** | ***MONTANT(FCFA)*** |
| *01* | *Ingénieur des travaux* | *14* | *10.000* | *140.000* |
| ***TOTAL*** | | | ***140.000*** | |

Tableau 5 :Ressources matérielles

|  |  |  |  |
| --- | --- | --- | --- |
| ***NOMBRE*** | ***DESIGNATION*** | ***PRIX UNITAIRE*** | ***MONTANT (F CFA)*** |
| *01* | *ordinateur de bureau de basse gamme pour le deploiement du client* | *90.000* | *90.000* |
| *01* | *ordinateur de bureau de moyenne gamme pour installer le serveur de BD* | *120.000* | *120.000* |
| ***TOTAL*** | | ***210.000*** | |

Tableau 6 :Ressources logicielles

|  |  |  |  |
| --- | --- | --- | --- |
| ***NOMBRE*** | ***DESIGNATION*** | ***PRIX UNITAIRE*** | ***MONTANT (F CFA)*** |
| *01* | *Système d’exploitation : Linux* | *Gratuit* | *-* |
| ***TOTAL*** | | | ***0*** |

Tableau 7 :Budget total du projet

|  |  |
| --- | --- |
| **DESIGNATIONS** | **MONTANT (F CFA)** |
| Ressources humaines | **140.000** |
| Ressources matérielles | **210.000** |
| Ressources logicielles | **0** |
| Risque du projet (10%) | **35000** |
| **TOTAL** | **385000** |

|  |
| --- |
| **Notre application nécessite 385000 FCFA** |

#### Organisation du projet

* + La phase d’analyse
  + La vérification et la validation de l’analyse
  + La conception des interfaces
  + La réalisation de l’application
  + Test et validation de l’application
  + Correction et validation
  + Déploiement et formation

#### Les livrables

* **Le programme d’installationt de application**
* **Le rapport complet :** 
  + Dossier d’analyse

## METHODOLOGIE

### Présentation de 2TUP

2TUP « **2 Track Unified Process** » encore prononcé « toutiyoupi », est un processus de développement logiciel qui implémente le processus unifié. Chacune des étapes du cycle découle des précédentes.

**Caractristiques :**

* + **Itératif:** Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et développer une solution effective de façon incrémentale par des itérations multiples.
  + **Piloté par les risques*:*** toutes les causes pouvant influencées la bonne évolution d’un projet doivent être identifiés et levés en priorité.
  + **Conduit par les cas d’utilisation*:*** le processus est orienté par les besoins utilisateurs présentés par des cas d’utilisation.
  + **Orienté composant*:*** Un composant est un module indépendant, qui pourrait servir pour d’autres projets. c’est une garantie de souplesse pour le logiciel à réutiliser.

La méthode 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle des deux branches à savoir : la branche fonctionnelle et la branche technique. La figure suivante est une illustration de la méthode 2TUP :

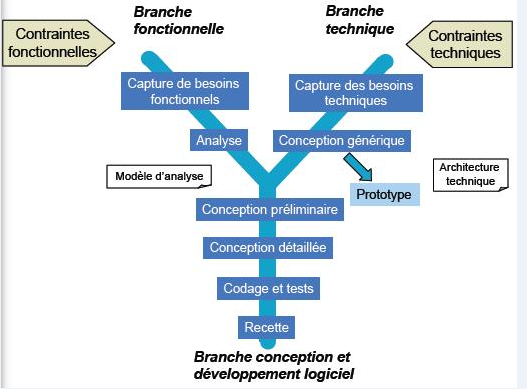


Figure 2: processus de développement 2TUP

### Relation entre 2TUP et d’UML



Figure 3: relation entre 2TUP et UML

## MODELISATION DE LA SOLUTION

### CAPTURE DES BESOINS FONCTIONNELS

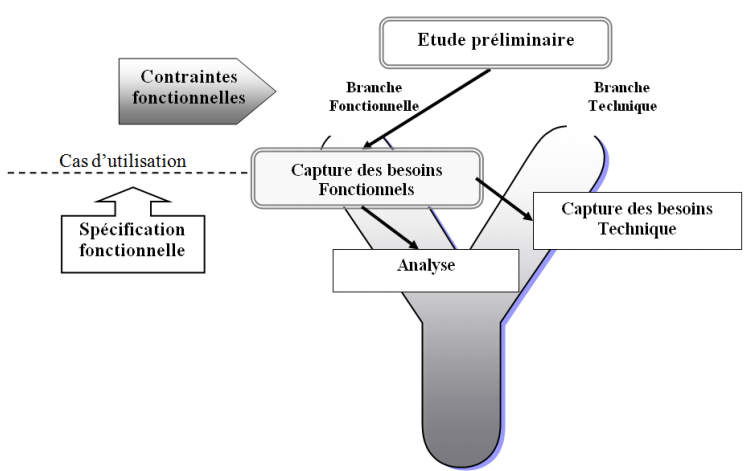


Figure 3: étude préliminaire

Les besoins fonctionnelles sont a revoir dans le cahier de charge

#### Diagrammes de cas d’utilisation

Un cas d'utilisation (Use case) «représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier».

Le diagramme des cas d’utilisation comporte les éléments suivants :

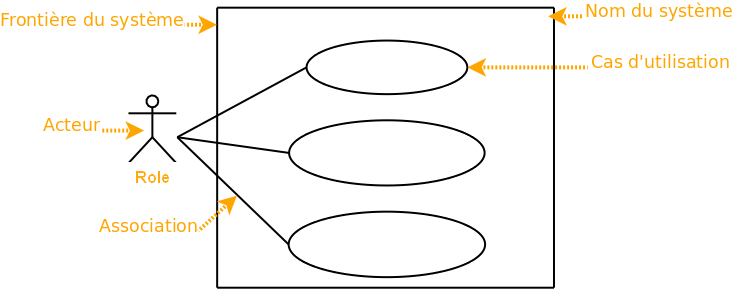
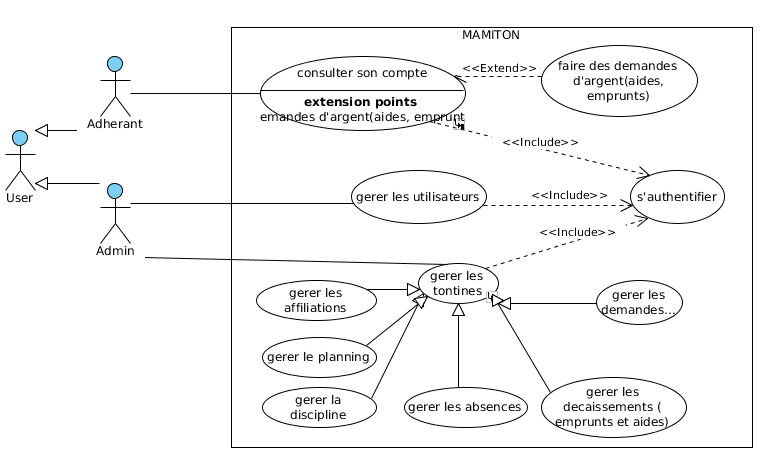


Figure 4:formalisme des use case

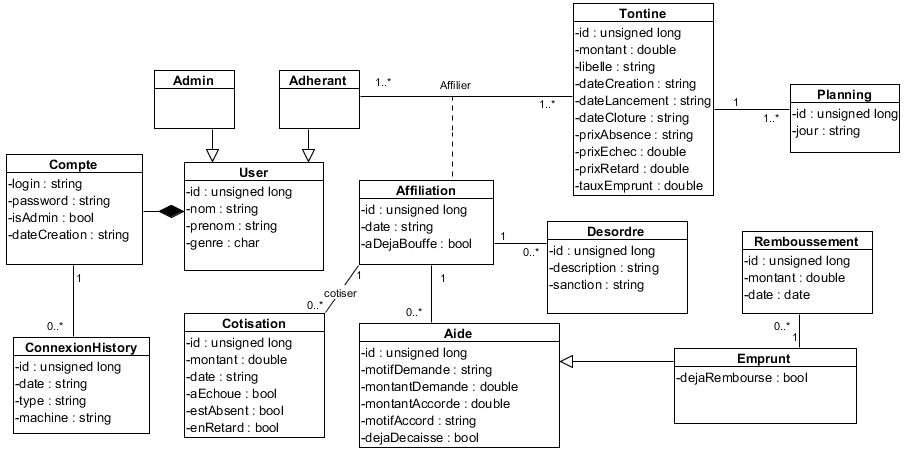
Diagramme de cas d’utilisation obtenus :



### ANALYSE ET CONCEPTION

#### Diagramme de classe de conception

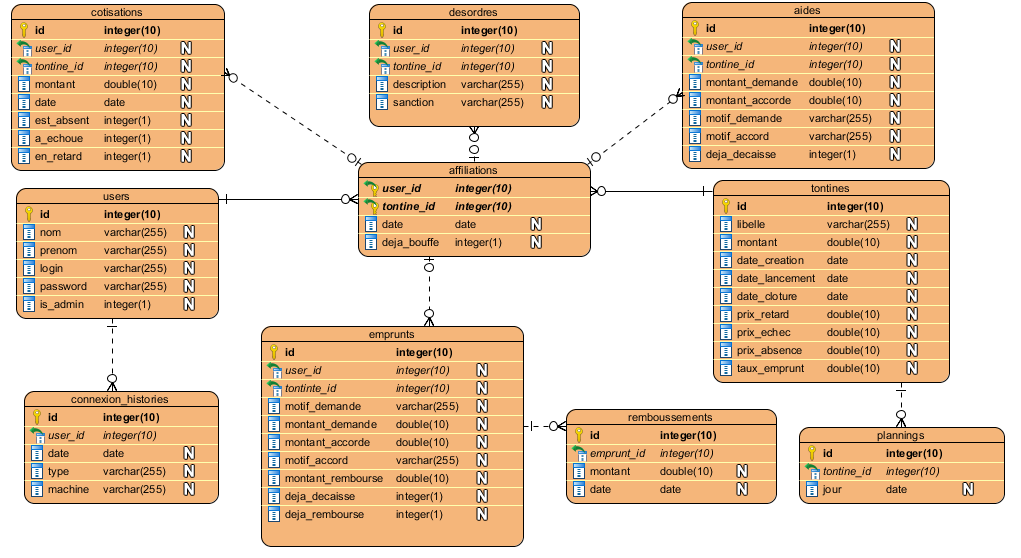
Figure 4:diagramme class de conception



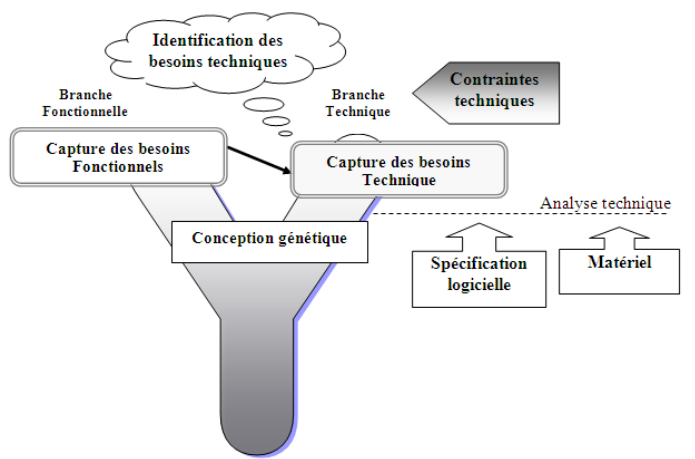
#### ERD (Entity Relationship Diagram)

Est un modele equivalent a MDP recommander pour les methodes d’analyse objet et les langages de modelisation objet

Figure : ERD :Modele Physique de donnees



### CONCEPTION GENERIQUE

On va s’intéresser à la branche droite du cycle en Y qui est « la capture des besoins techniques » en couvrant avec celle des besoins fonctionnels les contraintes qui ne traitent pas la description applicative.

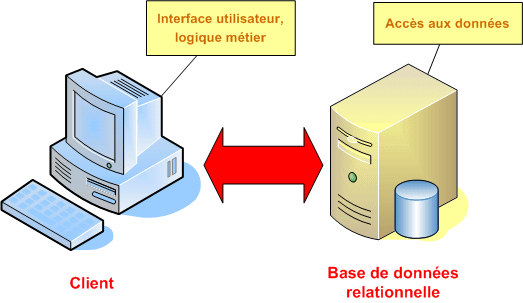
#### Architectures de l’application

Définition :

L’architecture est l’ensemble des décisions d’organisation du système logiciel qui défend les intérêts de son propriétaire final. Les intérêts en termes d’exigences fonctionnelles, techniques et économiques. L’architecture y répond par l’intégration de plusieurs styles de développement informatique qu’elle adapte aux éléments logiciels d’un contexte existant.

##### Présentation de l’architecture 2-tier

Notre application web sera déployée sur une architecture 2 tiers, cette architecture est décrite dans par la figure suivante.



#### Diagramme de deploiement

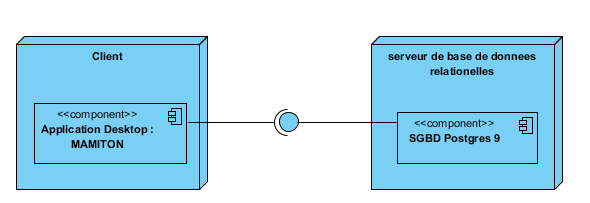


Table des matières

[DEDICACE ii](#_Toc476724083)

[REMERCIEMENTS iii](#_Toc476724084)

[LISTE DES FIGURES iv](#_Toc476724085)

[LISTE DES TABLEAUX v](#_Toc476724086)

[RESUME vi](#_Toc476724087)

[INTRODUCTION GENERALE 7](#_Toc476724088)

[CHAPITRE I: DOSSIER D’ANALYSE 9](#_Toc476724089)

[**INTRODUCTION** 9](#_Toc476724090)

[**I.** **Contexte** 10](#_Toc476724091)

[**II.** **Problématique** 10](#_Toc476724092)

[**III.** **Solutions probables** 11](#_Toc476724093)

[**IV.** **La Cible** 11](#_Toc476724094)

[**V.** **Impacts de la solution** 11](#_Toc476724095)

[**VI.** **Cahier de charge** 12](#_Toc476724096)

[1. Objectifs General 12](#_Toc476724097)

[2. Exigences 12](#_Toc476724098)

[3. Objectifs spécifique 12](#_Toc476724099)

[4. Déroulement du projet 14](#_Toc476724100)

[**VII.** **METHODOLOGIE** 16](#_Toc476724101)

[1. Présentation de 2TUP 16](#_Toc476724102)

[2. Relation entre 2TUP et d’UML 17](#_Toc476724103)

[**VIII.** **MODELISATION DE LA SOLUTION** 18](#_Toc476724104)

[**A.** **CAPTURE DES BESOINS FONCTIONNELS** 18](#_Toc476724105)

[**B.** **ANALYSE ET CONCEPTION** 20](#_Toc476724106)

[**C.** **CONCEPTION GENERIQUE** 22](#_Toc476724107)