

RAPPORT D'ANALYSE

ENTREPRISE DE VENTE : MIRACLE INFORMATIQUE SARL

Boutin François – Burkard Lucas – Mosser Olivier – Simeu Stéphanie

TABLE DES MATIERES

Introduction.....	4
Lettre de mission.....	5
Organisme client.....	5
Identification	5
Interlocuteurs	5
Nature du projet.....	5
Type du projet.....	5
Origine et objectif du projet	6
Délais	6
Moyen mis à disposition.....	6
Budget	6
Réserve budgétaire pour aléas.....	6
Intervenants affectés au projet.....	6
Moyens matériels.....	6
Responsabilité – Autonomie	7
Authentification	7
Note de clarification	8
Contexte.....	8
Définition du projet et objectifs visés	8
Problématique	8
Objectifs visés.....	8
Début du projet.....	8
Fin du projet	8
Budget du projet	9
Acteurs du projet.....	9
Maître d’ouvrage.....	9
Maître d’oeuvre	9
Compétences nécessaires.....	9
Produit du projet	10
Risques.....	10
Contraintes à respecter.....	10

Contraintes de délais.....	10
Contraintes de coût.....	10
Contraintes de performances	10
Authentification	11
Organigramme produit.....	12
Environnants du produit du projet.....	12
Fonctions.....	12
Classement des fonctions	12
Gestion des approvisionnements et stock.....	12
Gestion des achats et ventes	13
Gestion service après vente.....	13
Produit du projet	14
Organigramme.....	15
Diagramme de Gant.....	15
Analyse des risques	16
Modélisation.....	17
Modèle Conceptuel de Données	17
Modèle Relationnel.....	18
CONCLUSION.....	18

INTRODUCTION

Nous avons tiré au sort le sujet numéro 3 qui concerne la réalisation d'une base de données pour l'entreprise de vente Miracle Informatique SARL. Nous allons au cours de ce document, essayer de clarifier le sujet, le conceptualiser et le modéliser de façon à le rendre le plus compréhensible possible afin de faciliter son implémentation par l'équipe qui en aura la charge. Ainsi, nous

CLARIFICATION

LETTRE DE MISSION

ORGANISME CLIENT

IDENTIFICATION

La société pour qui nous allons effectuer ce projet se nomme Miracle Informatique SARL et est localisé 20 rue Roger Couttolenc à Compiègne.

INTERLOCUTEURS

Nos deux interlocuteurs principaux seront

- M. Stéphane Crozat, le maître d'ouvrage et Président Directeur Général de la
- Société Miracle Informatique SARL.
- M. Antoine Vincent, Directeur Technique et représentant principal de la société au sein de notre projet.

-

Leurs coordonnées respectives sont :

Stéphane Crozat : stephane.crozat@utc.fr

Antoine Vincent : antoine.vincent@utc.fr

NATURE DU PROJET

TYPE DU PROJET

Notre projet consiste à réaliser une base de données qui va permettre à l'entreprise de vente, Miracle Informatique SARL de gérer ses ventes, achats et améliorer son service après-vente. Pour se faire, nous devons premièrement concevoir la base de données et ensuite implémenter l'interface web en langage xhtml/css, php et postgresql afin qu'il puisse permettre une utilisation facile et ludique pour l'utilisateur ainsi que permettre de gérer au mieux les ventes et achats.

ORIGINE ET OBJECTIF DU PROJET

Le produit final sera une base de données modifiable à partir d'une interface web. Il se devra donc d'être de bonne qualité pour l'entreprise cliente afin que ceux-ci puissent améliorer leur service après-vente, gérer les ventes, l'approvisionnement et stock, et les relations avec les fournisseurs.

DELAIS

Le projet doit être terminé le Mercredi 18 Juin 2014.

MOYEN MIS A DISPOSITION

BUDGET

Le projet étant réalisé par un groupe d'étudiant dans le cadre d'une UV, le budget est de ce fait nul.

RESERVE BUDGETAIRE POUR ALEAS

Aucune

INTERVENANTS AFFECTES AU PROJET

Nous serons quatre à intervenir sur ce projet :

- Stéphanie Simeu : Chef de projet
- François Boutin : Responsable Qualité
- Lucas Burkard : Responsable Technique
- Olivier Mosser : Responsable Modélisation

MOYENS MATERIELS

Nous disposons des locaux de l'Université de Technologie de Compiègne qui dispose de tout un parc informatique ainsi que de tous les logiciels nécessaires :

- Réseau UTC Capella
- PostgreSQL
- Outil de modélisation UML

RESPONSABILITE – AUTONOMIE

Le chef de projet travaille sous la direction sa supérieure hiérarchique Mr Antoine Vincent, avec qui il fera le point chaque semaine sur l'avancement du projet. Pour le reste, des réunions régulières (une par semaine) de l'équipe seront organisés.

C'est au chef de projet qu'il incombe la responsabilité de répartir les tâches et de coordonner le travail des différents intervenants. Il est responsable du bon avancement du projet.

AUTHENTIFICATION

Date : 19/04/2014

Signature du Chef de projet

Stéphanie Simeu

Signature du Maître d'ouvrage

Stéphane Crozat

NOTE DE CLARIFICATION

CONTEXTE

L'Université de Technologie de Compiègne est une école d'ingénieur et une université réputée à l'échelle internationale. C'est dans le cadre de l'UV NF17, que les responsables de cet UV ont mobilisés une équipe pour la conception et réalisation d'une nouvelle base de données au profit de Miracle Informatique SARL, une entreprise de vente de matériel technologique et électroménager.

Miracle Informatique SARL étant une entreprise fictive représentée par les responsables et intervenants de l'UV NF17, elle est cependant considérée comme l'organisme client dans le cadre de ce projet.

DEFINITION DU PROJET ET OBJECTIFS VISES

PROBLEMATIQUE

Dans le cadre de cette simulation, Miracle Informatique SARL souhaite que l'on réalise une nouvelle base de données, qui va lui permettre de tracer précisément tous les achats effectués par les clients, afin d'améliorer tout son service après-vente.

OBJECTIFS VISES

Développer la base données qui devra permettre à l'entreprise de :

- Gérer les achats effectués par les clients,
- Connaître les informations sur ses produits, leur état (panne, irrécupérable,...)
- Le nombre de produits les plus vendus, le panier moyen
- Connaître le total des remises effectuées par un vendeur
- Gérer les approvisionnements et les stocks
- Gérer les relations avec les fournisseurs

DEBUT DU PROJET

Le projet a officiellement débuté le 26/02/2014, au début du semestre de printemps 2014.

FIN DU PROJET

Le projet doit être livré avant la fin du semestre de printemps 2014. Autrement dit, le mercredi 18 juin 2014.

BUDGET DU PROJET

- le projet se déroule sur 3 mois et 3 semaines à raison d'une séance hebdomadaire de 1 heure pendant 15 séances.
- La charge horaire totale du projet est de 60 heures réparties sur 4 personnes, soit une charge de 15h de travail par personne.
- Une salle machine équipée d'au minimum 13 ordinateurs sous Linux, un serveur sous Unix avec les logiciels nécessaires qui sont mis à notre disposition par le client et ne représentent donc aucun coût.

ACTEURS DU PROJET

MAITRE D'OUVRAGE

La maîtrise d'ouvrage est sous la responsabilité de :

- Stéphane Crozat : Président Directeur Général de Miracle Informatique SARL
- Antoine Vincent : Directeur Technique

MAITRE D'ŒUVRE

Le maître d'œuvre est le chef de projet, Stéphanie Simeu, qui a pour rôle de répartir les tâches et de planifier la réalisation du projet. Lors de la conception du système d'information, le maître d'œuvre aura un devoir de conseil vis-à-vis du maître d'ouvrage, car le système informatique doit tirer le meilleur parti des possibilités techniques.

COMPETENCES NECESSAIRES

- Gestion de projet : planification, définition des objectifs, répartition des tâches, contacts avec l'entreprise cliente (chef de projet).
- Compétences de modélisation : (Standard UML), normes sur la modélisation, connaissances du langage SQL (responsable modélisation).
- Implémentation de la base de données : Maîtrise des outils PostgreSQL 8 et PHP 5, implémenter le modèle préétabli (responsable technique).
- Test de qualité : tester le bon fonctionnement de la base de données et des interfaces à travers de tests d'utilisation les plus exhaustifs possibles afin de déceler toute erreur possible. En cas de mauvaise manipulation du matériel informatique par le client, ce sera alors au responsable technique de résoudre le problème et d'effectuer auparavant des tests pour vérifier le fonctionnement matériel.

PRODUIT DU PROJET

Le produit du projet est une base de données qui va permettre à l'entreprise de gérer ses clients, recenser le nombre de commandes effectuées par rapport à un produit vendu et les informations sur la clientèle, le produit le plus vendu.

Cette base de données devra être modifiable à partir d'une interface web : on pourra consulter les informations, les insérer, les modifier, faire une mise à jour, en rajouter, en supprimer...

Elle sera accessible par les employés membre du service vente, qui auront des droits d'accès assez réduit contrairement à l'administrateur, qui est aussi un employé de l'entreprise pouvant modifier, ajouter ou supprimer des informations sur un produit...

Les livrables du projet sont :

- Le MCD, rendu intermédiaire prévu le mercredi 26 mars 2014
- Un rapport d'analyse doit être livré le 16 avril 2014.
- La soutenance aura lieu le 18 juin 2014.

RISQUES

- Mauvaise modélisation du projet entraînant une base de données de moindre qualité.
- La base de données ne correspond pas à tous les critères demandés.
- Modèle conceptuel redondant.
- Manque d'expérience de la part d'un ou de plusieurs membres de l'équipe projet maîtrise d'œuvre, au niveau de la compréhension de conception d'une base de donnée proprement dite.

CONTRAINTES A RESPECTER

CONTRAINTES DE DELAIS

- Réunion de validation pour le MCD : 26 mars 2014.
- Réunion du rendu de rapport d'analyse : 16 avril 2014.

CONTRAINTES DE COUT

Aucune contrainte de coût.

CONTRAINTES DE PERFORMANCES

Le livrable souhaité doit fonctionner sur le matériel informatique de l'UTC.

AUTHENTIFICATION

Date : 12/03/2014

Signature du Chef de Projet : Stéphanie Simeu

Version 1.0

Signature du Maître d'ouvrage : Stéphane Crozat

ORGANIGRAMME PRODUIT

ENVIRONNANTS DU PRODUIT DU PROJET

- Administrateur ou Gérant
- Employés du service vente
- Employés du service achat

FONCTIONS

Le produit du projet doit permettre aux Employés du service vente de :

- Consulter les produits et connaître le stock
- Gérer les clients
- Établir des facturations : soit lors de la vente, lors de la réparation, lors de la reprise d'un objet et imprimer.
- Connaître les fournisseurs

Le produit du projet doit permettre aux Gérants ou Administrateurs de :

- Gérer les produits : Ajouter, supprimer, modifier un produit et faire la mise à jour
- Consulter les produits en stocks

CLASSEMENT DES FONCTIONS

GESTION DES APPROVISIONNEMENTS ET STOCK

Pour satisfaire les besoins du service vente, le produit du projet doit leur permettre :

- Consulter les produits : avoir toutes les informations sur un produit
- Connaître le stock des produits en vente

Pour satisfaire le besoin du gérant ou administrateur, le produit du projet doit lui permettre de :

- Consulter les produits et connaître leur stock
- Gérer les produits : Ajout, Modification, Suppression

GESTION DES ACHATS ET VENTES

Pour satisfaire les besoins du service vente, le produit du projet doit leur permettre :

- Gérer les clients
- Établir les factures suite aux ventes des produits
- Gérer les ventes : informations sur toutes les ventes effectuées

Pour satisfaire les besoins du service achat, le produit du projet doit leur permettre :

- Effectuer les bons de commande
- Connaître le stock des produits

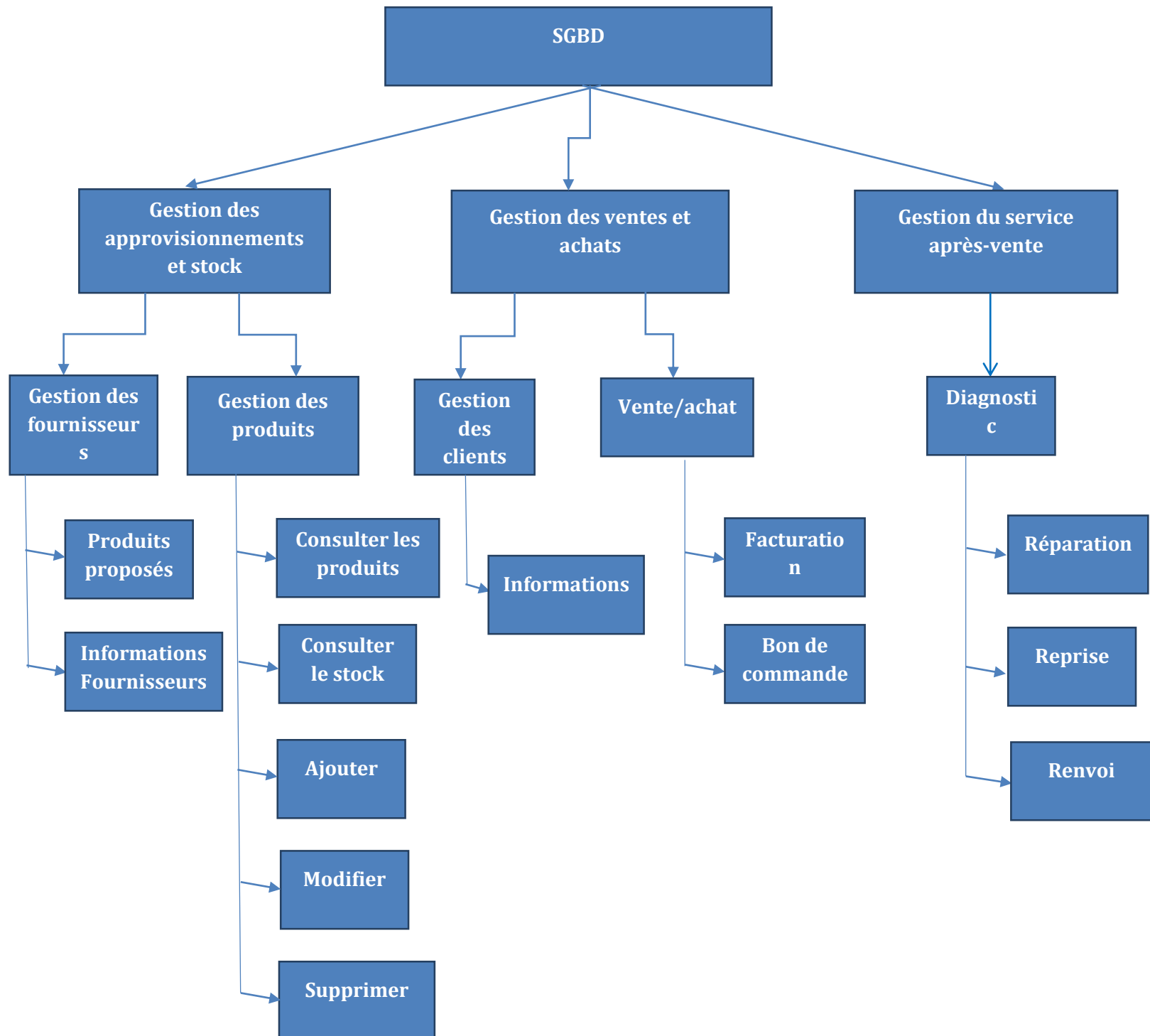
GESTION SERVICE APRES-VENTE

Le service après-vente s'occupe du diagnostic des produits. Ainsi, il peut soit réparer le produit, soit le renvoyer au client et soit faire une reprise sur ce dernier.

Le produit du projet permettra donc de savoir :

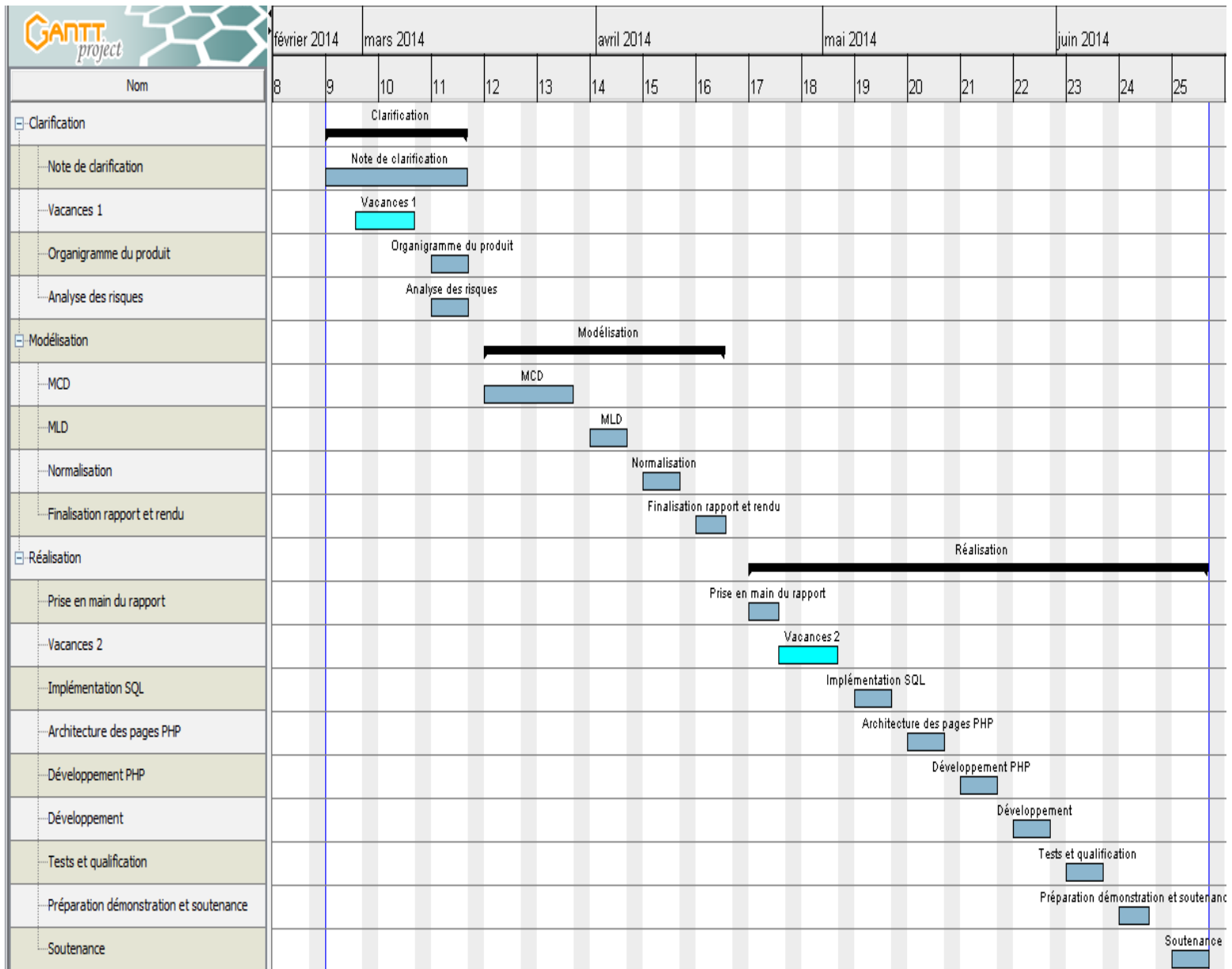
- Le temps passé sur la réparation d'un produit
- L'état d'un produit
- Le matériel utilisé pour réparer un produit

PRODUIT DU PROJET



ORGANIGRAMME

DIAGRAMME DE GANT

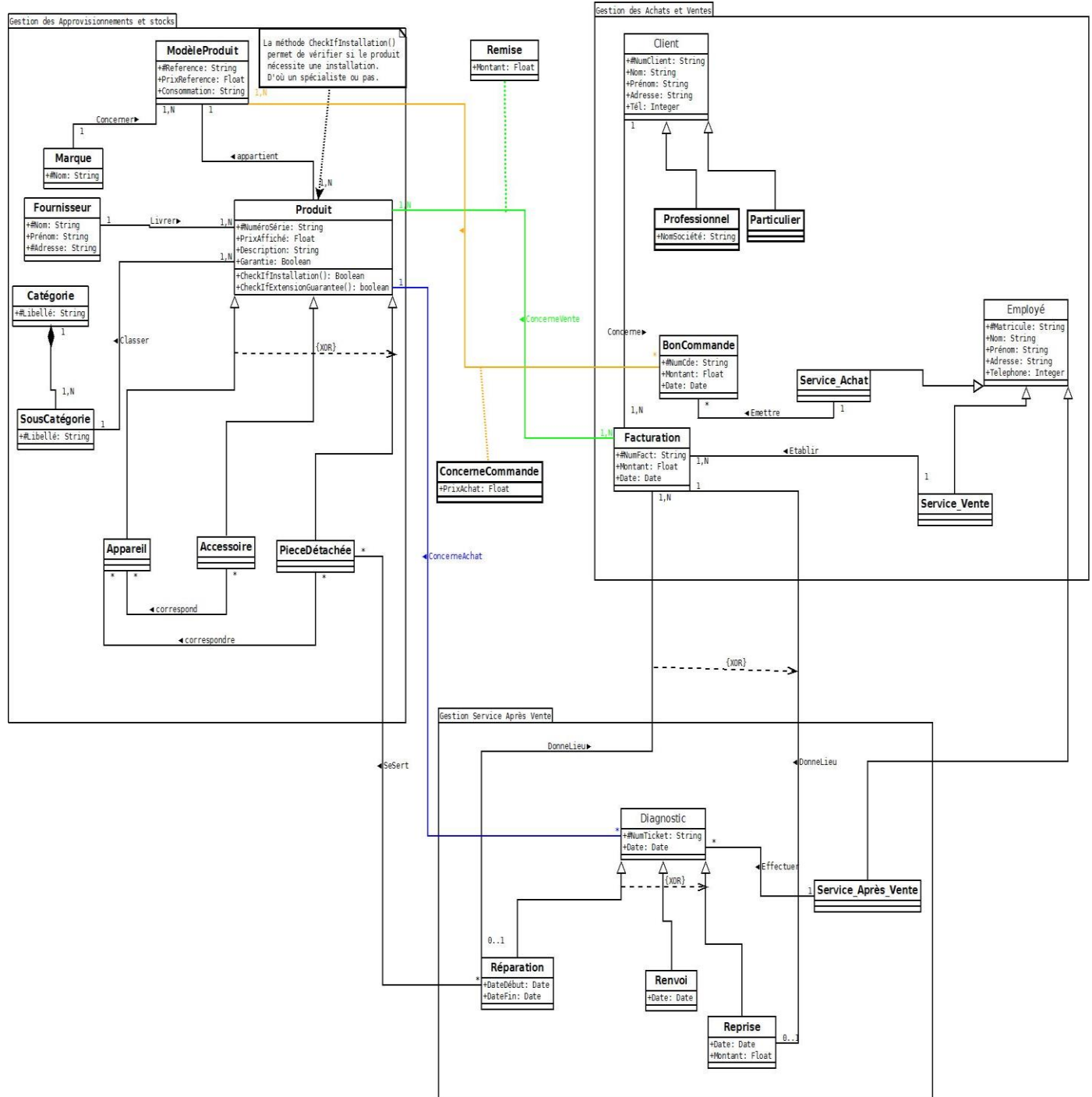


ANALYSE DES RISQUES

DEFINITION ET ANALYSE DES RISQUES							
N°	Description	Causes	Conséquences	Criticité			
				Gravité	Fréquence	Détection	Criticité
1	Mauvaise modélisation de la base de données	Incompréhension du sujet	Sujet trop large ou au contraire trop étroit entraînant une non réponse au problème initial posé par le client	2	3	2	12
		Mauvaises hypothèses faites					
		Incompatibilité entre les classes et/ou modules	Problèmes lors de l'implémentation	2	2	1	4
		mauvaise définition des fonctions à implémenter	Solution ne permettant pas d'effectuer les requêtes demandées initialement par le client	2	3	2	12
2	Retard du lancement		Dépassement du temps de réalisation	3	3	1	9
3	Les compétences des ressources ne sont pas au niveau requis		Qualité déficiente du produit	3	2	2	12
4	Analyse des besoins incomplets	Produit ne répondant pas aux attentes du client	Qualité déficiente du produit	4	2	2	16
5	Urgences liées à un autre projet	Défaut de ressource	Dépassement des temps de réalisation	2	2	1	4
6	Défaillance de la communication	réalisation non adéquate au projet	Dépassement de temps de réalisation Dépenses supplémentaires	3	2	3	18
7	Mauvaise modélisation de la base de données relationnelle	Problème de normalisation	Problèmes pour l'implémentation	3	3	2	18

MODELISATION

MODELE CONCEPTUEL DE DONNEES



MODÈLE RELATIONNEL

ModèleProduit (#Reference: string, PrixReference: float, Consommation:string, Marque : string =>Marque(Nom))

Fournisseur (#Nom : string, #Adresse : string, Prénom : string)

Marque (#Nom : string)

Catégorie (#Libellé : string)

SousCatégorie (#Libellé : string, #catégorie : string =>Catégorie(Libellé))

Produit (#NuméroSérie : string, PrixAffiché : float, Description : string, Garantie : boolean, NomFournisseur : string =>Fournisseur, AdresseFournisseur : string =>Fournisseur, SousCatégorie : string =>SousCatégorie(Libellé), Modèle:string=>ModèleProduit(Référence),

Avec Marque, NomFournisseur, AdresseFournisseur, Catégorie, Modèle, NuméroFacturation
NOT NULL

Contraintes :

- Proj (Fournisseur, Nom, Adresse) IN Proj (Produit, NomFournisseur, AdresseFournisseur)
- Proj (Catégorie, Libellé) IN Proj (Produit, Catégorie)
- Proj (Marque, Nom) IN Proj (Produit, Marque)
- Proj (ModèleProduit, Référence) IN Proj (Produit, Modèle)
- Proj (Facturation, NumFact) IN Proj (ConcerneVente, NuméroFacturation)

Héritage non exclusif, héritage pas complet et classe mère non abstraite : héritage par référence

vAppareil= jointure (Type, Appareil, CodeType=CodeAppareil)

vAccessoire= jointure (Type, Accessoire, CodeType=CodeAccessoire)

vPieceDétachée= jointure (Type, PieceDétachée, CodeType=CodePieceDétachée)

PieceDétachée (#CodePieceDétachée : string =>Type(CodeType))

Appareil (#CodeType : string =>Type(CodeType))

Accessoire (#CodeAccessoire : string =>Type(CodeType))

Correspond (#CodeAppareil : string =>Appareil(CodeType), #CodeAccessoire : string
=>Accessoire(CodeType))

Correspondre (#CodeAppareil : string =>Appareil(CodeType), #CodePieceDétachée : string
=>PieceDétachée(CodeType))

Contrainte:

Client : classe mère abstraite, héritage exclusif et héritage presque complet. Ainsi, nous choisissons l'héritage par la classe mère.

Client (#NumClient: string, Nom: string, Prénom: string, Adresse: string, Tel: integer, RaisonSociale, Type: {Prof|Part})

Contrainte:

vProfessionnel= Projection (Restriction (Client, Type=Prof), NumClient, Nom, Prénom, Adresse, Tél, RaisonSociale)

vParticulier= Projection (Restriction (Client, Type=Part), NumClient, Nom, Prénom, Adresse, Tél)

Employé : classe mère abstraite et relation avec service d'où héritage par référence pour faciliter le processus.

Employé (#Matricule: string, Nom: string, Prénom: string, Adresse: string, Tel: integer)

Contrainte :

-PROJ (Service,Nom) IN PROJ (Employé,NomService)

-PROJ (Employé,Matricule) IN (PROJ (Service_Achat,MatriculeAchat) UNION PROJ (Service_Vente, MatriculeVente) UNION PROJ (Service_Après_Vente, MatriculeAprèsVente))

Service_Achat (#MatriculeAchat=>Employé)

Service_Vente (#MatriculeVente=>Employé)

Service_Après_Vente (#MatriculeAprèsVente=>Employé)

VService_Achat=jointure (Employé, Service_Achat, Matricule=MatriculeAchat)

VService_Vente=jointure (Employé, Service_Vente, Matricule=MatriculeVente)

VService_Après_Vente=jointure (Employé, Service_Après_Vente, Matricule=MatriculeAprèsVente)

ConcerneVente (#NuméroFacturation: string =>Facturation(NumFact), Montant: float #Numsérie: string=>Produit(NumSérie))

Contrainte :

-PROJ (Produit, NumSérie) IN PROJ (ConcerneVente, NumSérie) AND

-PROJ (Facturation, NumCde) IN PROJ (ConcerneVente, NumCde)

Facturation (#NumFact : string, Montant : float, Date : date, NumClient : string=>Client(NumClient), Matricule : string=> Service_Vente(Matricule), NumTicket : string=>Reprise (NumTicket), NumTicket : string=>Reparation (NumTicket))

Avec NumClient, Matricule NOT NULL and NumTicket est soit celui de la Reprise ou celui de la réparation mais pas les deux à la fois ;

Contrainte :

-PROJ (Client, NumClient) IN PROJ (Facturation, NumClient)

-PROJ (Service_Vente,Matricule) IN PROJ (Facturation,Matricule)

-PROJ (Reparation, NumTicket) IN PROJ (Facturation, NumTicket)

BonCommande (#NumCde : string, Montant : float, Date : date, Matricule: string
=>Service_Achat(Matricule))

Avec Matricule *NOT NULL*

ConcerneCommande (#Reference: string =>ModèleProduit (Reference), #NumCde: string
=>BonCommande (NumCde))

Contrainte :

- PROJ (ModèleProduit, Reference) IN PROJ (ConcerneCommande, Reference) AND
- PROJ (BonCommande, Numcde) IN PROJ (ConcerneCommande, NumCde)

Diagnostic : classe mère abstraite et relation avec produit et service_après-vente d'où héritage par référence car plus facile à implémenter.

Diagnostic (#NumTicket: string, Date: date, NumSérie: String=>Produit (NumSérie),
Matricule: string => Service_Après_Vente(Matricule))

Avec Matricule, NumSérie *NOT NULL*

SeSert ((#CodePieceDétachée: string=>PieceDétachée (CodeType), #NumTicket:
string=>Diagnostic (NumTicket))

Contrainte :

- PROJ (PieceDétachée, CodeType) IN PROJ (SeSert, CodeType) AND
- PROJ (Réparation, NumTicket) IN PROJ (SeSert, NumTicket)

Réparation (#NumTicketRéparation=>Diagnostic: string, DateDébut: date, DateFin: date)

Renvoi (#NumTicketRenvoi: string =>Diagnostic, Date: date)

Reprise ((#NumTicketReprise: string =>Diagnostic, Date: date, Montant: Float)

VRéparation=jointure (Diagnostic, Réparation, NumTicket=NumticketRéparation, Montant)

VRenvoi=jointure (Diagnostic, Renvoi, NumTicket=NumticketRenvoi)

VReprise=jointure (Diagnostic, Reprise, NumTicket=NumticketReprise, Montant)

Contrainte :

-PROJ (Diagnostic, NumTicket) IN (PROJ (Réparation, NumTicket) UNION PROJ (Renvoi, NumTicket) UNION PROJ (Reprise, NumTicket))

-XOR : Exactement un des trois (Réparation, Reprise, renvoi) devra être instancié après un diagnostic

NORMALISATION

Package des approvisionnements et stocks :

- **Produit :**
NuméroSérie -> PrixAffiché, Description, Garantie, Consommation, Type, NomFournisseur, AdresseFournisseur, SousCatégorie, Modèle

Fermeture transitive: Normes 2NF et 3NF respectées
- **ModèleProduit**

Reference -> PrixReference, Marque, Consommation

Fermeture transitive: Normes 2NF et 3NF respectées
- **Fournisseur**

Nom, Adresse -> Prénom

Fermeture transitive: Normes 2NF et 3NF respectées

Ces classes ci-dessous ont une clé unique donc 3NF respectée

- **Catégorie** (#Libellé : string)
- **Marque** (#Nom : string)
- **Appareil** (#CodeType : string =>Type(CodeType))
- **Accessoire** (#CodeAccessoire : string =>Type(CodeType))
- **SousCatégorie**

Libellé -> catégorie

Fermeture transitive : 2NF et 3NF respectées
- **PieceDétachée**

CodePieceDétachée -> NumTicket

Fermeture transitive : 2NF et 3NF respectées

Ces 3 relations ci-dessous sont logiquement en 3NF puisque formées uniquement de clés primaires

- **Correspond** (#CodeAppareil : string =>Appareil(CodeType), #CodeAccessoire : string =>Accessoire(CodeType))
- **Correspondre** (#CodeAppareil : string =>Appareil(CodeType), #CodePieceDétachée : string =>PieceDétachée(CodeType))
- **SeSert** ((#CodePieceDétachée: string=>PieceDétachée (CodeType), #NumTicket: string=>Diagnostic (NumTicket))

Package Service après-vente :

- DF :
 - Table Réparation :
NumTicketRéparation -> DateDébut
NumTicketRéparation -> DateFin
 - Table Renvoi :
NumTicketRenvoi -> Date
 - Table Reprise :
NumTicketReprise -> Date
NumTicketReprise -> Montant
 - Diagnostic
NumTicket-> Date
NumTicket-> NuméroSérie
NumTicket-> Matricule
- Forme normale du package :

Ce package est bien en troisième forme normale car :

 - Tous les attributs sont atomiques et chaque table possède une clé : 1NF.
 - Il n'y a pas de partie de clé donnant un attribut (car les clés sont toujours composées d'un seul attribut) : 2NF.
 - Il n'y a pas d'interdépendance(s) entre des attributs non-clés (seule la table Réparation aurait pu être dans ce cas car elle est la seule à posséder 2 attributs non-clés ; mais la date de début ne donne pas la date de fin et vice-versa) : 3NF.

Package Achats et ventes :

- DF :
- Table Client :
 - NumClient -> Nom
 - NumClient -> Prénom
 - NumClient -> Adresse
 - NumClient -> Tel
 - NumClient -> Type
- Table Employé :
 - Matricule -> Nom
 - Matricule -> Prénom
 - Matricule -> Adresse
 - Matricule -> Téléphone
- Ces classes ci-dessous ont une clé unique donc 3NF respectée
 - Service_Achat** (#MatriculeAchat=>Employé)
 - Service_Vente** (#MatriculeVente=>Employé)
 - Service_Après_Vente** (#MatriculeAprèsVente=>Employé)
- Table Facturation :
 - NumFact -> Montant, Date, NumClient, Matricule, NumTicket
 - NB: le NumTicket peut être soit celui du Renvoi, soit celui de la réparation et pas les 2 à la fois.**
- **Ces 3 relations ci-dessous sont logiquement en 3NF puisque formées uniquement de clés primaires**
 - Concerne** (#NuméroSérie: string =>Produit (NuméroSérie), #NumCde: string =>BonCommande (NumCde))
 - BonCommande** (#NumCde : string, Montant : float, Date : date, Matricule: string =>Service_Achat(Matricule))
 - ConcerneVente** (#NuméroFacturation: string=>Facturation(NumFact), Montant:float, #Numsérie:string=>Produit(NumSérie))

- Forme normale du package :

Ce package est bien en troisième forme normale car :

- Tous les attributs sont atomiques et chaque table possède une clé : 1NF.
- Il n'y a pas de partie de clé donnant un attribut (car les clés sont toujours composées d'un seul attribut) : 2NF.
- Il n'y a pas d'interdépendance(s) entre des attributs non-clés (exemple : Le prénom du client ne donne pas son nom et vice-versa) : 3NF.

CONCLUSION

Le projet de réalisation d'une base de données pour une Entreprise de vente de matériel électroménager et informatique est un projet ambitieux. Nous avons essayé lors de cette première partie de modéliser les besoins du client de la façon la plus claire qu'il soit. Le temps dédié à la deuxième partie étant assez limitée, nous pensons que lors de la réalisation logicielle de ce projet il faudra faire des choix sur ce qu'il faut implémenter complètement, partiellement ou pas du tout. Nous souhaitons bonne chance à l'équipe qui s'occupera de la suite de ce projet et espérons que ce document couvrira au mieux son objectif qui est de clarifier le sujet et de rendre ainsi son implémentation la plus facile et la plus adaptée possible.