

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université ABDERRAHMANE MIRA –Bejaïa-

Faculté des Sciences Exactes

Département Informatique



Mémoire de fin de cycle

En vue d'obtenir le diplôme de master professionnel en Génie logiciel

Thème

*Développement Mobile Multiplateforme d'une application de mise en relation d'affaires sous l'AGL WINDEV MOBILE :
Modélisation, conception des bases de données*

Réalisé par :

M^r. MERROUDJ Akli

M^r. SIDI Ali Abdellatif

Jury:

Président : M^r. ALLICHE Abdennour

Examinatrice : M^{lle}. AZOUI Aicha

Promoteur : Dr. IDOUGHI Djilali

Remerciements

Au terme de notre travail, qu'il nous soit permis d'exprimer notre profonde gratitude :

- A DIEU Tout Puissant, pour nous avoir donné la force, la santé et l'intelligence nécessaire pour accomplir ce travail ;

- A nos parents et tous nos frères et sœurs pour leurs volontés, leurs énormes sacrifices et pour l'amour qu'ils ont su nous donner tout au long de notre formation. Nous leur sommes infiniment reconnaissants ;

- A tout le corps professoral et administratif du département informatique de l'université de Bejaia pour l'ensemble des efforts, des enseignements et conseils prodigués à notre égard ;

- A notre encadreur, Mr le Docteur IDOUGHI Djilali. Pour nous avoir soutenu, guidé, encouragé et orienté en prenant à cœur ce travail malgré ses charges horaires et multiples occupations, Nous lui sommes très reconnaissants ;

-A toute personne qui a contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

***SIDI ALI Abdellatif** : Je dédie ce travail à mes chers parents pour leur soutien moral et leur encouragement. À ma sœur Samira, mon petit frère Younes. À tous mes proches sans exception. À mon binôme Tarik, À Seddam et Nassim et à tous mes amis (Ferroudj Adel, yahia, Kiki, wali, Rachid, Brahim). À ma future femme Farah. Et surtout À ma chère nanna Ourida paix a son âme.*

***MERROUDJ Akli** : Je dédie ce modeste travail d'abord à mes très chers Parents. et aussi pour mes frères et sœurs et la femme de mon frère et son fils, Et surtout À mon chère papa Idir paix a son âme. sans oublier ma future femme Lyfia. Je le dédie aussi pour mes proches, À mon binôme Batiche. À Seddam et Nassim et mes copains de chambre et à tous mes amis (Wali, Yaya, Brahim, Adel, Salim, Nabil, Sofiane, Sadek,.....etc).*

Table des matières

Remerciements	2
Dédicaces	Erreur ! Signet non défini.
Table des matières	4
Liste des figures	7
Liste des tableaux	10
Chapitre I : Génie logiciel:Généralités	Erreur ! Signet non défini.
Introduction	Erreur ! Signet non défini.
1.Généralités	Erreur ! Signet non défini.
2.Cycle de vie d'un Logiciel	Erreur ! Signet non défini.
3.Les principaux modèles de cycle de vie	Erreur ! Signet non défini.
4.Qu'est ce qu'un atelier de génie logiciel?	Erreur ! Signet non défini.
Conclusion	Erreur ! Signet non défini.
Chapitre II : Développement mobile multiplateforme	Erreur ! Signet non défini.
Introduction	Erreur ! Signet non défini.
Partie1 : Introduction à la mobilité	Erreur ! Signet non défini.
1.Qu'est ce qu'une application mobile ?	Erreur ! Signet non défini.
2.Les principales plateformes mobiles	Erreur ! Signet non défini.
3.Généralités sur le développement mobile	Erreur ! Signet non défini.
4.Les outils du développement mobile	Erreur ! Signet non défini.
Partie 2 : Présentation de L'AGL Windev Mobile	Erreur ! Signet non défini.
1.Généralités sur Windev mobile	Erreur ! Signet non défini.
2.Fonctionnalités de Windev mobile	Erreur ! Signet non défini.
3.Cycle de développement d'une application mobile	Erreur ! Signet non défini.
4.Centre de suivi de projets	Erreur ! Signet non défini.
5.La structure d'un projet Windev	Erreur ! Signet non défini.
6.Architecture d'une application Windev Mobile	Erreur ! Signet non défini.
7. WLangage	Erreur ! Signet non défini.
7.1.Définition	Erreur ! Signet non défini.
7.2.Les variables	Erreur ! Signet non défini.
7.3.Les constantes	Erreur ! Signet non défini.
7.4.Les instructions de base	Erreur ! Signet non défini.
7.5.Les fonctions	Erreur ! Signet non défini.

7.6. Les modes de programmation	Erreur ! Signet non défini.
Partie 3 : Bases de données dans Windev mobile	Erreur ! Signet non défini.
1. Généralités	Erreur ! Signet non défini.
2. Structure d'une base de données.....	Erreur ! Signet non défini.
3. Cycle de création d'une base de données	Erreur ! Signet non défini.
4. Bases de données tierces sous Windev mobile.....	Erreur ! Signet non défini.
5. Base de données HFSQL.....	Erreur ! Signet non défini.
5.1. Présentation	Erreur ! Signet non défini.
5.2. Les avantages de HFSQL.....	Erreur ! Signet non défini.
5.3. Les caractéristiques techniques de HFSQL	Erreur ! Signet non défini.
5.4. Les modes de fonctionnement d'une application HFSQL	Erreur ! Signet non défini.
5.4.1. HFSQL Classic	Erreur ! Signet non défini.
5.4.2. HFSQL client/serveur	Erreur ! Signet non défini.
5.4.2.1. Outils de HFSQL	Erreur ! Signet non défini.
5.4.2.2. Fonctions HFSQL.....	Erreur ! Signet non défini.
5.4.2.3. Centre de contrôle HFSQL.....	Erreur ! Signet non défini.
6. Les requêtes.....	Erreur ! Signet non défini.
6.1. Types de requêtes HFSQL.....	Erreur ! Signet non défini.
6.2. Création de requêtes	Erreur ! Signet non défini.
6.3. Editeur de requêtes	Erreur ! Signet non défini.
Conclusion.....	Erreur ! Signet non défini.
Chapitre III : Etude de cas- modélisation et conception de la base de donnée de l'application yperesia	Erreur ! Signet non défini.
Introduction	Erreur ! Signet non défini.
1. Cadre de notre projet.....	Erreur ! Signet non défini.
2. Démarche adoptée	Erreur ! Signet non défini.
3. Description du contexte de l'application	Erreur ! Signet non défini.
4. Acteurs de la base de données	Erreur ! Signet non défini.
5. Modélisation et conception de la base de données	Erreur ! Signet non défini.
5.1. Les niveaux de modélisation d'une base de données	Erreur ! Signet non défini.
5.2. Méthodologie.....	Erreur ! Signet non défini.
5.3. Présentation du diagramme de classes d'analyse	Erreur ! Signet non défini.
5.4. Dictionnaire des données	Erreur ! Signet non défini.
5.5. Le modèle conceptuel des données.....	Erreur ! Signet non défini.
5.6. Le modèle logique des données.....	Erreur ! Signet non défini.
5.7. Code SQL correspondant à la création de la base de données	Erreur ! Signet non défini.

Conclusion	Erreur ! Signet non défini.
Chapitre IV : Realisation et Evaluation	Erreur ! Signet non défini.
Introduction	Erreur ! Signet non défini.
1.Matériel utilisé	Erreur ! Signet non défini.
2.Fonctionnement de la base de données	Erreur ! Signet non défini.
3.Captures d'écran relatives aux différentes opérations sur la base de données	Erreur ! Signet non défini.
4.Test de la base de données	Erreur ! Signet non défini.
Conclusion	Erreur ! Signet non défini.
Perspectives	Erreur ! Signet non défini.
Glossaire	Erreur ! Signet non défini.
Liste des abréviations	Erreur ! Signet non défini.
Références bibliographiques	Erreur ! Signet non défini.
Annexe	102

Liste des figures

Figure 1 : les besoins de développement.....	13
Figure 2 : cycle de vie d'un logiciel.....	15
Figure 3 : Modèle en cascade.....	16
Figure 4 : modèle en v.....	16
Figure 5 : modèle itératif et incrémental.....	17
Figure 6 : cycle de développement d'une application sous Windev mobile..	28
Figure 7 : le centre de suivi de projet.....	29
Figure 8 : architecture d'une application mobile.....	31
Figure 9 : structure d'un projet Windev mobile.....	36
Figure 10 : exemple d'une base de données.....	38
Figure 11 : importation d'une base de données.....	39
Figure 12 : base de données accessible par Windev mobile et modes d'accès.....	41
Figure 13 : manipulation d'une base de données HFSQL mobile.....	45
Figure 14 : fonctionnement d'une application client/serveur.....	46
Figure 15 : exemple d'une requête de sélection.....	50
Figure 16 : Code SQL de la requête sélection.....	50
Figure 17 : exemple d'une requête d'insertion.....	50
Figure 18 : Code SQL de la requête insertion.....	50

Figure 19 : exemple d'une requête modification.....	52
Figure 20 : code SQL de la requête modification.....	52
Figure 21 : exemple d'une requête suppression.....	53
Figure 22 : code SQL de la requête suppression.....	53
Figure 23 : exemple d'une requête composé.....	54
Figure 24 : code SQL de la requête composé.....	55
Figure 25 : Démarche adopté.....	58
Figure 26: les différentes version de notre application.....	58
Figure 27 : les acteurs qui interagisse avec la base de données.....	59
Figure 28 : Niveau de modélisation d'une base de données.....	61
Figure 29: Méthodologie de modélisation.....	63
Figure 30 : Diagramme de classe d'analyse.....	64
Figure 31 : exemple de transformation de classe.....	67
Figure 32 : exemple de transformation d'héritage.....	69
Figure 33 : model conceptuel des données de la base de données.....	70
Figure 34 :model logique des données généré par Windev mobile.....	74
Figure 35: code SQL pour la création de fichier de données client.....	75
Figure 36 : code SQL pour la création de fichier de données personne_morale.....	75
Figure 37 : code SQL pour la création de fichier de données personne_physique.....	76
Figure 38 : code SQL pour la création de fichier de données poste.....	76
Figure 39 : code SQL pour la création de fichier de données publication.....	76
Figure 40 : code SQL pour la création de fichier de données recherche.....	77

Figure 41 : code SQL pour la création de fichier de données recherche1.....	77
Figure 42 : fonctionnement de la base de données de l'application yperesia.	81
Figure 43 : fenêtre correspondante à la vérification.....	82
Figure 44 : code WLangage qui implémente la verification d'un enregistrement.....	83
Figure 45 : fenêtre correspondante a la mise a jour des informations de client.....	84
Figure 46 : code WLangage de la mise a jour des informations de client.....	85
Figure 47 : fenêtre correspondante à la recherche d'un fournisseur.....	86
Figure 48 : fenêtre correspondante à l'insertion d'un enregistrement dans la table personne_physique.....	87
Figure 49: code WLangage pour l'insertion d'un enregistrement dans la table client.....	88
Figure 50 : fenêtre correspondante à la mise a jour de la valeur de champ mot_de_passe.....	88
Figure 51 : code WLangage pour la verification de champ mot_de_passe.....	89
Figure 52 :Code WLangage pour connecter la base de données yperesia au serveur HFSQL.....	89
Figure 53 : test d'une requete de modification.....	90

Liste des tableaux

Tableau 1 :tableau récapitulatif des outils et fonctionnalités de Windev mobile.....	25
Tableau 2 : tableau récapitulatif des éléments d'un projet Windev mobile.....	30
Tableau 3:Vocabulaire de Windev mobile liée au concept de base de données	36
Tableau 4:fonctions essentiels pour manipuler les bases de données tiers...39	
Tableau 5 :caractéristiques technique de HFSQL.....	42
Tableau 6 :les fonctions essentiels de HFSQL.....	47
Tableau 7 :répartition des taches.....	57
Tableau 8 :vision globale des interaction Acteur/base de données.....	60
Tableau 9 :les niveaux de modélisation et les model associe.....	62
Tableau 10 :dictionnaire de données	67

Introduction générale

Les bases de données sont utilisées dans de nombreuses applications et intégrées dans notre paysage et dans nos usages quotidiens pour nous donner accès à de nouveaux services

Elles ont pris aujourd'hui une place essentielle dans l'informatique, et des concepts et des méthodes ont été développées pour gérer leur complexité. C'est une discipline qui s'appuie sur une théorie solide et qui offre de nombreuses techniques pour la modélisation et la conception.

Une personne est souvent confronté à chercher un service qui est proposé par une autre personne. Pour réussir des échanges profitables entre le client et le fournisseur du service il est nécessaire de faciliter la communication. Un client doit facilement avoir les informations sur le service proposé et sur celui qui le fournit, et le fournisseur doit connaître ses clients et faciliter l'accès à son service.

C'est dans ce cadre que nous proposons une solution simple et efficace qui est la conception d'une base de données pour une application mobile afin d'assurer une meilleure mise en relation des fournisseurs et des clients.

Nous répartissons notre travail en quatre chapitres :

Le premier chapitre intitulé "Généralités sur le génie logiciel" porte sur la présentation du génie logiciel et l'exposition de ses aspects essentiels.

Le deuxième chapitre intitulé "Développement mobile multiplateforme" aborde dans sa première partie le développement mobile et présente l'environnement Windev Mobile, et dans sa deuxième partie il détaille la conception des bases de données dans cet environnement.

Le troisième chapitre intitulé "étude de cas : modélisation et conception de la base de données d'une application mobile pour la mise en relation de services" porte sur la modélisation et la conception de la base de données.

Le quatrième chapitre intitulé "Réalisation et évaluation" porte sur la réalisation finale et les tests de la base de données.

CHAPITRE I :
Génie logiciel : Généralités

CHAPITRE 1

Génie logiciel - Généralités

Introduction

Dans ce chapitre, nous allons présenter les concepts essentiels et les outils pour la bonne conduite d'un projet de développement informatique dans le cadre du génie logiciel.

1. Généralités

Selon L'IEEE 1990, le génie logiciel est l'application une approche systématique, disciplinée et quantifiable pour le développement, l'opération et la maintenance de logiciels.

L'objectif du génie logiciel est de maîtriser le cout et le délai et de produire des logiciels cohérents et fiables.

Le logiciel est un objet immatériel et complexe, Ses caractéristiques attendues sont difficiles à figer au départ et souvent remises en cause en cours de développement. Sa production est une activité créative, mais qui doit se conduire avec une certaine rigueur.

Pour réussir un projet il faut décomposer le développement d'un logiciel en un processus ayant des étapes plus gérables et décrire le logiciel selon plusieurs perspectives (*modèles*), chacune mettant l'emphase sur un aspect particulière. La figure1 montre les différents éléments pour la réussite d'un projet de développement logiciel [1].

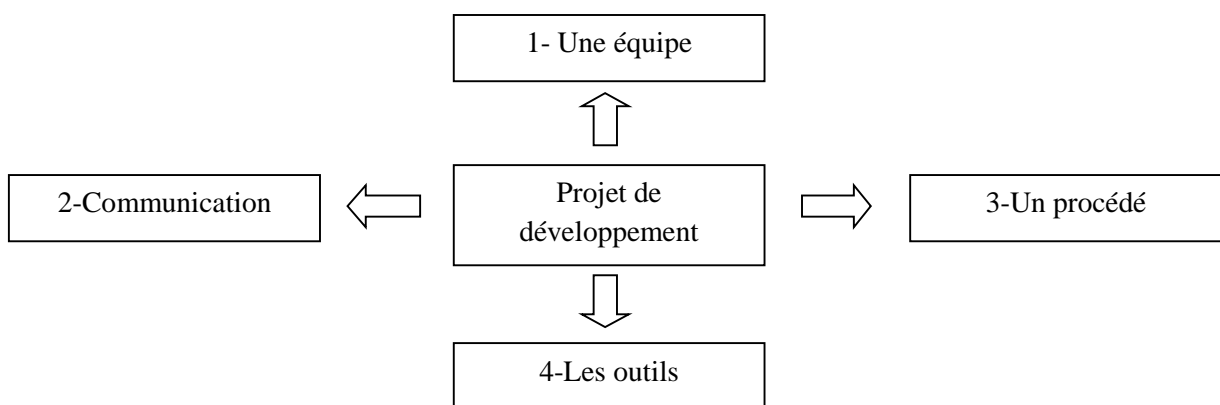


Figure 1 : Les besoins du développement

Pour réussir un projet de développement logiciel il faut réunir quatre éléments essentiels : une équipe, une bonne communication, un procédé adapté et les bons outils.

CHAPITRE 1

Génie logiciel - Généralités

1- Equipe

Le développement d'un logiciel requiert une équipe regroupant différentes compétences au delà de l'écriture du code et une organisation permettant à ces compétences de s'exercer dans les conditions optimales.

2- Communication

L'équipe de développement logiciel doit adopter et s'adapter aux bonnes pratiques qui incitent à la communication et à la circulation d'informations dans le but d'accroître la productivité et permettre un bon suivi du projet.

3- Procédé

Le procédé de développement regroupe, organise et automatise l'ensemble des étapes à suivre pour produire le logiciel final et détermine l'ensemble des documents et des livrables tout au long du processus.

4- Outils

Les outils de développement regroupent l'ensemble des outils matériels et logiciels pour organiser, structurer, suivre et gérer le projet logiciel, le travail en équipe [2].

2. Cycle de vie d'un Logiciel

Un Cycle de développement logiciel définit un ensemble d'activités et leur enchaînement. Une activité comprend : des tâches, des contraintes, des ressources et une façon d'être réalisée.

L'IEEE [ANSI/IEEE 1002] propose un modèle général en cinq phases pour les cycles de vie comme le montre la figure 2.

CHAPITRE 1

Génie logiciel - Généralités

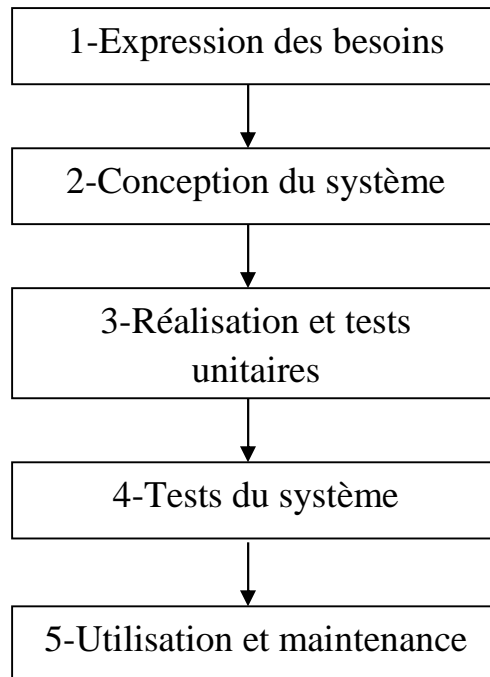


Figure 2 : Cycle de vie d'un Logiciel

- 1- Expression des besoins :** consiste à consulter les utilisateurs, définir les fonctionnalités du système et rédiger des documents compréhensibles par les utilisateurs et les équipes de développement.
- 2- Conception du système et du logiciel :** L'activité de conception regroupe le recensement des diverses fonctions et la décomposition du système en architectures logicielles et matérielles.
- 3- Réalisation et tests unitaires :** cette étape se focalise sur le choix d'un langage de programmation et la production des programmes exécutables et puis sur la génération de tests unitaires pour ces programmes.
- 4- Tests du système :** regroupe l'intégration des unités du programme, les tests de l'ensemble et la livraison aux utilisateurs.
- 5- Utilisation et maintenance :** regroupe la correction des erreurs, l'Amélioration des programmes, l'augmentation des fonctionnalités au fur et à mesure des besoins et la remise en cause des étapes précédentes [3].

3. Les principaux modèles de cycle de vie

Il existe un certain nombre de modèles de cycle de vie et la plupart de ces modèles des processus reprennent les activités fondamentales mais les organisent différemment. De nombreux modèles ont été définis ; un modèle peut être spécifique à une organisation et à un type de logiciels. On cite entre autres :

1- Modèle en cascade

Le modèle en cascade considère le développement logiciel comme une succession d'étapes réalisées de façon strictement séquentielle. Chaque étape correspond à une activité de base et elle doit être validée.

Le processus de développement utilisant un cycle en cascade exécute des phases qui ont pour caractéristiques :

- de produire des livrables définis au préalable.
- de se terminer à une date précise.
- de ne se terminer que lorsque les livrables sont jugés satisfaisants lors d'une étape de validation-vérification. [4]

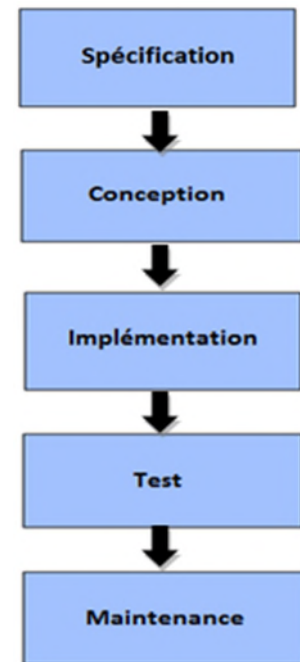


Figure 3 : Modèle en cascade

2- Modèle en V

Ce modèle est une amélioration du modèle en cascade qui permet en cas d'anomalie, de limiter un retour aux étapes précédentes. Les phases de la partie montante doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés afin d'améliorer le logiciel.

Issu du monde de l'industrie, le cycle en V est devenu un standard de l'industrie logicielle depuis les années 1980 [4].

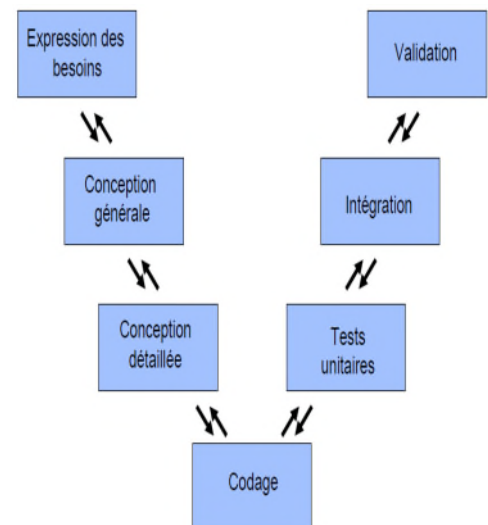


Figure 4 : Modèle en V

3- Modèle itératif et incrémental

Ce modèle se base sur le découpage fonctionnel en sous ensembles qui seront développés par incréments. Pour chaque incrément on repasse par toutes les étapes.

Le développement est essentiellement basé sur l'analyse des résultats pour faire évoluer le produit progressivement [4].

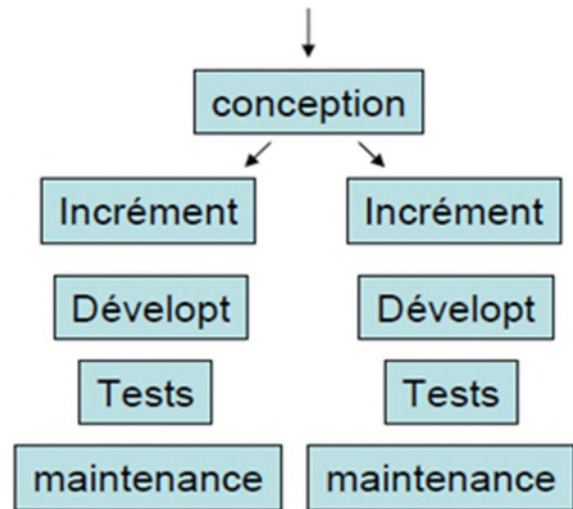


Figure 5 : Modèle itératif et incrémental

4. Qu'est ce qu'un atelier de génie logiciel?

Un AGL (Atelier de Génie Logiciel) ou atelier CASE (Computer Aided Software Engineering) est un environnement de développement logiciel aidant à la réalisation de logiciels. Autrement dit, il s'agit d'un système pour le développement logiciel assisté par ordinateur. Un AGL intègre des outils adaptés aux différentes phases de la production d'un logiciel et facilite la communication et la coordination entre ces différentes phases. Un AGL est basé sur des méthodologies qui formalisent le processus logiciel, et à l'intérieur de ce processus, chacune des phases qui le composent.

Les AGL apportent une réelle solution à certains problèmes du génie logiciel et contribuent nettement à l'amélioration de la productivité et de la qualité du logiciel, notamment en faisant le suivi des différentes phases du processus logiciel et en offrant un cadre cohérent et uniforme de production. Néanmoins, cet enthousiasme doit être modéré: le processus logiciel est encore loin d'être maîtrisé et les différentes formalisations qui en sont proposées font encore l'objet de controverses, et dans tous les cas, sont bien loin d'être totalement automatisables.

On distingue essentiellement trois types d'AGL selon la nature des outils intégrés:

CHAPITRE 1

Génie logiciel - Généralités

1. Les environnements de conception (upper-case)

Ces ateliers s'intéressent plus particulièrement aux phases d'analyse et de conception du processus logiciel. Ils intègrent généralement des outils pour l'édition de diagrammes (avec vérification syntaxique), des dictionnaires de données, des outils pour l'édition de rapports, des générateurs de (squelettes de) code, des outils pour le prototypage, ... Ces ateliers sont généralement basés sur une méthode d'analyse et de conception et utilisés pour l'analyse et la conception des systèmes d'information.

On cite entre autres TRAMIS qui est un environnement de conception qui intègre notamment un éditeur de diagrammes et un générateur de prototypes.

2. Les environnements de développement (lower-case)

Ces ateliers s'intéressent plus particulièrement aux phases d'implémentation et de test du processus logiciel. Ils intègrent généralement des éditeurs (éventuellement dirigés par la syntaxe), des générateurs d'interfaces homme/machine, des SGBD, des compilateurs, optimiseurs, pretty-printers, debuggers, ...

On cite entre autres Windev.

WinDev est un environnement de développement.

Un autre exemple est Unix qui intègre différents outils pour la programmation et le test. L'intégration des données est faite par l'intermédiaire des fichiers Unix, la gestion (limitée) de configurations est faite par make...

Certains environnements, plus évolués, sont dédiés à un langage particulier. Il existe par exemple des environnements dédiés à Inter Lisp, Smalltalk, Loops... Ces différents environnements proposent des bibliothèques de composants, une interface graphique, des éditeurs dédiés au langage, des interprètes, debuggers, ... Ces environnements permettent un développement rapide et convivial. En revanche, l'application développée est intégrée dans l'environnement, ce qui peut poser des problèmes de portabilité et de coût [5].

CHAPITRE 1

Génie logiciel - Généralités

4- Les environnements de développement multiplateforme

Il existe des solutions pour permettre de ne développer qu'une fois l'application puis de la déployer sur d'autres plateformes.

Ces solutions sont proposées par différents éditeurs de logiciels et reposent sur différentes approches, chacune a ses avantages et ses inconvénients. Elles se basent généralement sur le principe "write once, run anywhere" ou "coder une fois et exécuter n'importe où".

Et parmi ces solutions nous citons : Titanium appcelerator, rhomobile, Adobe Flash Builder, MOBL, Open Plug et Windev Mobile [6].

Conclusion

Notre projet de PFE s'inscrit dans le cadre du génie logiciel, et consiste à développer la partie base de données d'un logiciel de type application mobile dans l'environnement multiplateforme Windev Mobile, que nous décrivons dans le chapitre 2.

CHAPITRE II :
Développement mobile
multiplateforme

Introduction

Le développement mobile est confronté à la diversité des plateformes et des technologies de développement. Windev Mobile se positionne comme l'une des meilleures solutions pour le développement multiplateforme.

La première partie de ce chapitre aborde la notion de mobilité et la deuxième est consacrée à une présentation rapide de l'environnement Windev mobile.

La dernière partie expose les outils proposés par Windev mobile pour la modélisation, la conception et l'implémentation des bases de données qui est l'objet principal de notre travail.

Partie1 : Introduction à la mobilité

1. Qu'est ce qu'une application mobile ?

Une application mobile est un logiciel applicatif développé pour un appareil électronique mobile, tel qu'un assistant personnel, un téléphone portable, un Smartphone, un baladeur numérique ou une tablette tactile.

Les applications mobiles sont pour la plupart distribuées depuis des plateformes de téléchargement (parfois elles-mêmes contrôlées par les fabricants de Smartphones) telles que l'App Store (plateforme d'Apple), le Google Play (plateforme de Google / Android), ou encore le Windows Phone Store (plateforme de Microsoft). Mais des applications peuvent aussi être installées sur un ordinateur, grâce par exemple au logiciel iTunes distribué par Apple pour ses appareils. Les applications distribuées à partir des magasins d'applications sont soit payantes, soit gratuites, mais généralement avec des publicités.

Sur certaines plateformes, les applications peuvent aussi être installées à partir de sources tierces, via un site non affilié au distributeur d'origine.

2. Les principales plateformes mobiles

Il existe une multitude de plateformes mobiles sur le marché, chacune possède ses forces et ses faiblesses, et est différemment apprécié des professionnels suivant les goûts et les couleurs de chacun.

CHAPITRE 2

Développement mobile multiplateforme

On cite entre autres :

IOS : est le système d'exploitation intégré à la fois dans l'iPhone, l'iPod Touch et l'iPad, et mis au point par Apple. IOS est dérivé de Mac OS X et donc d'Unix.

Android : est le système d'exploitation conçu par Google et l'Alliance. Celui-ci équipe aujourd'hui des appareils très variés : Smartphones, tablettes, TV, autoradios, montres digitales. Android est gratuit pour les constructeurs d'appareils souhaitant l'utiliser, et partiellement open-source.

Windows phone : lancé en novembre 2010 par Microsoft, il succède à Windows Mobile en étant plus orienté grand public. Il est basé sur un noyau Windows CE.

BlackBerry OS: est le système du fabricant canadien RIM, fut un précurseur sur le marché des PDA et Smartphones. Il fut le premier à proposer la notification instantanée d'emails, en mode push. Il optimise également l'utilisation mobile en compressant les pages web, ainsi que les pièces jointes des mails, et permet une sécurisation efficace des échanges. En septembre 2010, RIM a lancé un nouveau système d'exploitation, BlackBerry Tablet OS, qui équipe aujourd'hui uniquement sa tablette BlackBerry PlayBook, mais qui devrait à terme remplacer totalement son BlackBerry OS sous le nom de BlackBerry 10.

Bada : est le système d'exploitation pour mobiles mis au point par Samsung pour équiper une partie de ses Smartphones. Bada OS est une évolution pour le haut de gamme du système SHP (Samsung Handheld Platform) OS équipant de nombreux appareils Samsung de milieu de gamme. Tous les Smartphones Samsung équipés du système Bada ont un nom commençant par Samsung Wave.

Symbian OS : crée par Nokia en 1998 en compagnie de Panasonic, Psion, Ericsson et Motorola. Nokia fut ensuite le principal utilisateur de Symbian pendant de nombreuses années pour équiper ses téléphones mobiles et Smartphones, et racheta tous les droits du consortium Symbian Ltd en 2008.

Firefox OS : Anciennement nommé Boot To Gecko, Firefox OS est un système d'exploitation pour Smartphones et tablettes créé par Mozilla, et entièrement open-source. Il a été annoncé en 2011 et est toujours en cours de développement. L'objectif est de proposer un OS mobile reposant sur des standards ouverts du Web [7].

3. Généralités sur le développement mobile

L'univers du développement d'applications mobiles évolue de manière constante et se dirige vers une uniformisation des codes. En effet, de plus en plus d'environnements mobiles permettent de créer des applications sur la base d'un seul langage tout en permettant leur distribution sur des systèmes variés. Plusieurs moyens existent et offrent des avantages variés, suivant les besoins.

Il existe plusieurs techniques de développement : le développement hybride a aujourd'hui le vent en poupe. Cette méthode repose surtout sur l'utilisation de langage web. L'outil de développement hybride le plus connu est Phonegap, qui permet de compiler le code sous la forme d'une application. Ceci permet d'utiliser entièrement le potentiel du mobile et ses fonctionnalités.

Une autre technique consiste à encapsuler le code dans un conteneur natif, ce qui offre l'avantage de supprimer les intermédiaires entre le code et le système pour de meilleures performances. Une solution un peu différente consiste à mettre en place des applications sous la forme d'un site web (majoritairement HTML 5). On les nomme alors « webapps ». Elles offrent un accès simple via une adresse web [8].

4. Les outils du développement mobile

Les applications mobiles sont développées sur des ordinateurs ; le langage utilisé dépend du système sous lequel l'application sera exécutée et chaque plateforme possède ses propres outils de développement.

Ci-après nous allons présenter les outils de développement pour les trois principales plateformes mobiles : Android, IOS et Windows Phone.

Android : Pour développer une application mobile sur ce support, le langage de programmation utilisé sera Java. Le kit de développement proposé par Google : SDK (Software Development Kit) autrement dit les outils de développement

CHAPITRE 2

Développement mobile multiplateforme

utilisés seront le SDK Android. On peut également utiliser le langage C++ avec le NDK (Native Development Kit)

IOS : C'est la plate-forme Apple. Le langage utilisé sera l'Objective-c ou Swift. Pour développer des applications sur iOS il faut avoir un Mac, installer l'environnement de développement XCode et le avoir le Framework Cocoa Touch.

Windows Phone : Microsoft propose deux systèmes: un pour les tablettes et un pour les téléphones. Normalement ces derniers devraient fusionner d'ici la version 10 de Windows. Ici, le langage utilisé est généralement le C# accompagné des technologies web (HTML, CSS, JavaScript) via la technologie WinJS et l'environnement de développement utilisé est Microsoft Visual Studio.

Partie 2 : Présentation de L'AGL Windev Mobile

1. Généralités sur Windev mobile

Windev mobile est un AGL (Atelier de Génie Logiciel) édité par la société française PC SOFT [<http://www.pcsoft.fr>].

C'est un environnement de développement complet, intuitif et multiplateforme qui intègre tous les outils nécessaires au cycle de réalisation d'une application mobile.

Windev mobile est autonome, quelque soit le matériel cible et le système d'exploitation, la méthode de développement est similaire.

Il offre aussi aux développeurs un langage de cinquième Génération, le WLangage, qui est très simple et très puissant. Contrairement à d'autres langages de développement traditionnels, il n'est pas nécessaire de chercher et de rajouter des modules pour pouvoir concevoir, tester et installer une application mobile [9].

2. Fonctionnalités de Windev mobile

Windev Mobile est un environnement complet qui prend en charge tous les besoins d'un développement mobile. Il offre aux développeurs et aux chefs de projets un ensemble d'outils pour réaliser une application mobile.

CHAPITRE 2

Développement mobile multiplateforme

Ces outils sont répartis sur deux catégories : les centres de contrôle et les éditeurs et les fonctionnalités essentielles de Windev mobile sont :

- La gestion de l'équipe de développement.
- La gestion de la communication entre les différents intervenants du projet.
- La gestion du cycle de développement (conception, programmation, test, déploiement).
- La gestion des bases de données.
- La création des interfaces graphiques.

Le tableau 1 illustre l'ensemble des fonctionnalités et des outils proposés par Windev mobile.

Fonctionnalité	Outil
La gestion de l'équipe de développement	- Le centre de suivi de projets
La gestion de la communication entre les intervenants du projet	- Centre de gestion des documents et des ressources - Centre de contrôle qualité
La gestion du cycle de développement	- Editeur UML - Editeur de code - Editeur de modélisation souple
La gestion des bases de données	- Editeur d'analyse - Centre de contrôle HFSQL - Editeur de requêtes
La création des interfaces graphiques	- Editeur de fenêtres

Tableau 1 : Tableau récapitulatif des outils et fonctionnalités de Windev mobile

3. Cycle de développement d'une application mobile

Windev mobile est un environnement autonome. Il propose un cycle unifié pour le développement des applications mobiles.

Le développement sous Windev mobile est itératif et piloté par les tests, chaque module est développé et testé séparément. Il est basé sur quatre phases :

- **Conception** : constitue la définition des exigences issues du cahier de charges ou d'une étude du domaine de l'application puis la modélisation en utilisant les diagrammes UML ou le MCD de Merise.
- **Développement** : La première étape consiste à créer le projet et l'analyse qui correspond à la base de données et La deuxième à développer les différents modules (fenêtres, web services, classes, composants).

Dans cette phase il est possible de développer plusieurs modules en parallèle et les tester indépendamment.

- **Test et génération** : c'est la troisième phase du cycle de développement, elle consiste à tester l'application finale sur l'environnement Windev mobile puis générer le fichier qui sera installé et exécuté sur le terminal mobile.
- **Déploiement** : Le déploiement consiste à installer tous les fichiers nécessaires à la mise en marche de l'application mobile, soit l'exécutable ou les fichiers de données sur le serveur.

La Figure 6 illustre le cycle d'une application sur Windev mobile.

CHAPITRE 2

Développement mobile multiplateforme

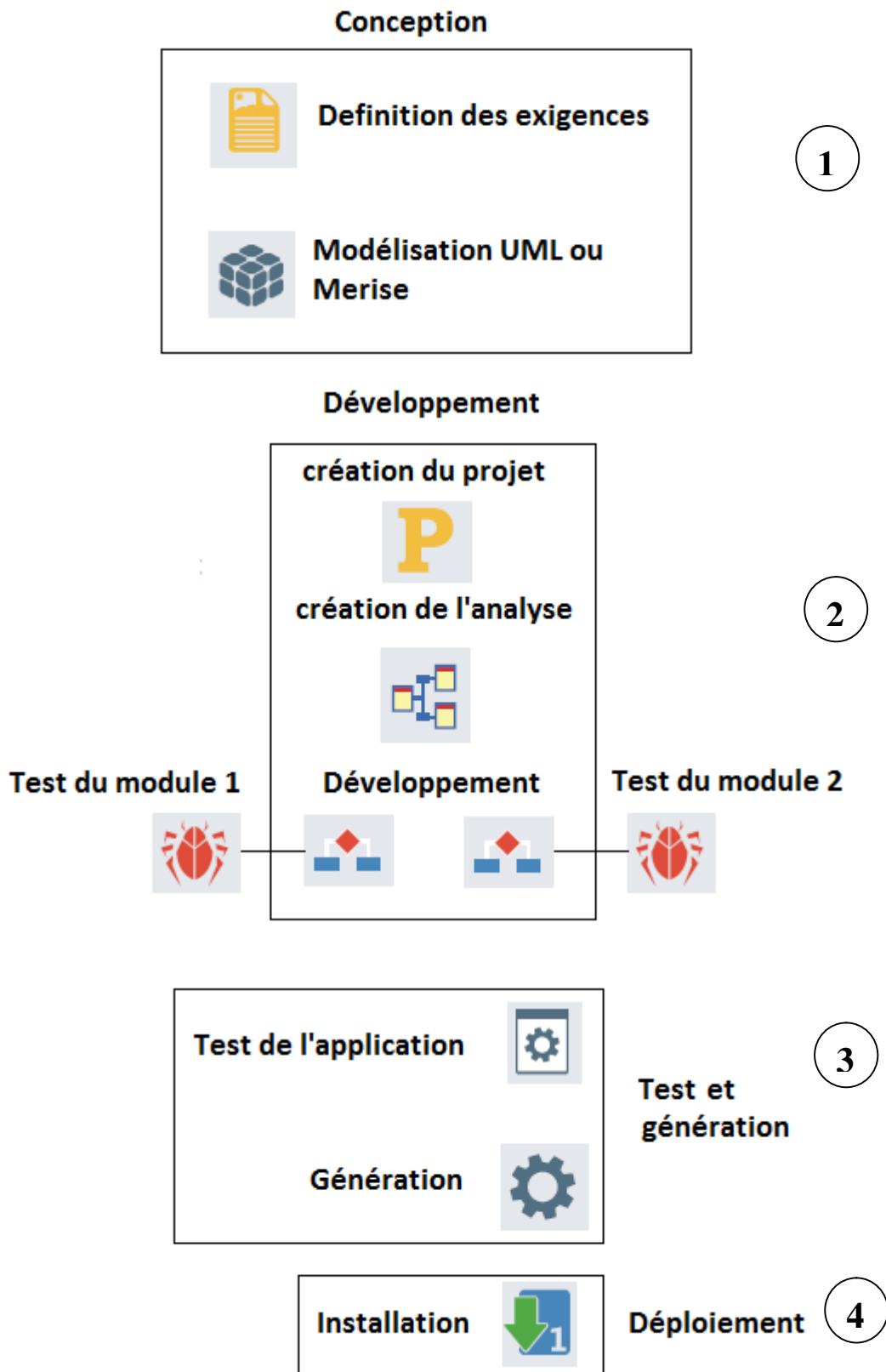


Figure 6 : Cycle de développement d'une application sous Windev mobile

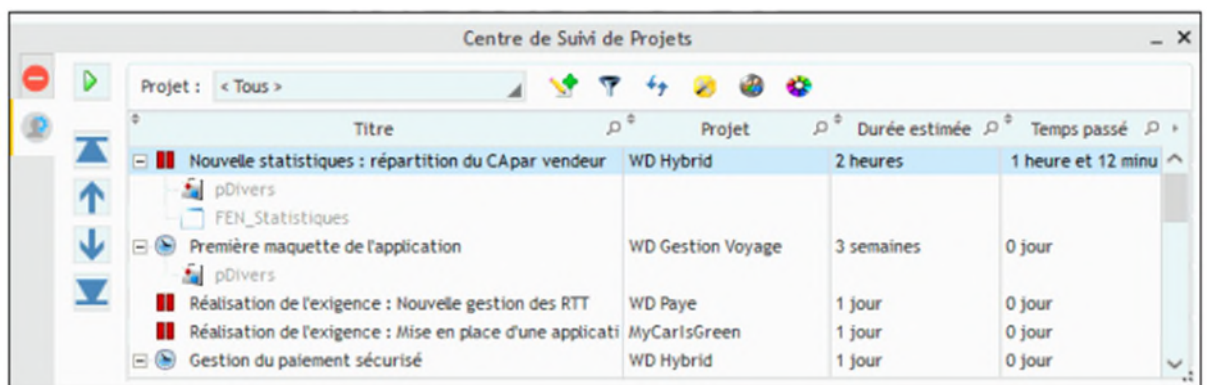
4. Centre de suivi de projets

Le Centre de Suivi de Projets permet de gérer le planning de réalisation d'un projet, de sa création à sa livraison finale. Il est ainsi possible de connaître l'ensemble des tâches à réaliser sur un projet, les développeurs concernés, les durées nécessaires, l'état d'avancement du projet, ...etc

Chaque développeur peut également visualiser directement sa liste de tâches grâce au "Centre de Suivi de Projets"

Ce centre de Suivi est utilisé par deux niveau :

- Utilisation par un administrateur (par défaut, le login et le mot de passe correspondent à "ADMIN").
L'administrateur définit les projets, les intervenants, les tâches à réaliser. Seul l'administrateur peut donner les droits d'administrateur ou supprimer une tâche, ..
- Utilisation par un intervenant :
L'intervenant peut voir son planning, ajouter des tâches, les déplacer, configurer ses horaires de travail, modifier l'état d'une tâche [10].



Titre	Projet	Durée estimée	Temps passé
Nouvelle statistiques : répartition du CA par vendeur	WD Hybrid	2 heures	1 heure et 12 minu
pDivers FEN_Statistiques			
Première maquette de l'application	WD Gestion Voyage	3 semaines	0 jour
pDivers			
Réalisation de l'exigence : Nouvelle gestion des RTT	WD Paye	1 jour	0 jour
Réalisation de l'exigence : Mise en place d'une applicati	MyCarIsGreen	1 jour	0 jour
Gestion du paiement sécurisé	WD Hybrid	1 jour	0 jour

Figure 7 : le centre de suivi de projet

5. La structure d'un projet Windev

Windev mobile permet de créer une application mobile grâce à un projet qui est enregistré sous la forme d'un fichier WPP.

CHAPITRE 2

Développement mobile multiplateforme

Un projet Windev mobile correspond à un ensemble d'éléments créés lors du développement. Ces éléments sont réutilisables ou partageables par d'autres projets.

Windev mobile propose deux outils pour la visualisation de la structure d'un projet : L'explorateur de projet et le graphe de projet.

Le tableau 2 illustre les différents éléments d'un projet.

Elément	présentation	Extension
Fenêtre	Interface graphique de l'application	WPW
Requête	Sert à interroger une base de données	WDR
Classe	Élément orienté objet qui représente une entité du projet ou du domaine du projet. Il est composé d'un ensemble de membre, de méthodes et de propriétés.	WDC
Modèle de champ	Élément regroupant un ensemble de champs réutilisables par plusieurs fenêtres, ainsi une modification du modèle est appliquée pour tous les champs qui l'utilisent.	WDT
Modèle de fenêtre	Élément qui contient tous les éléments graphiques commun à un ensemble de fenêtres, ainsi une modification du modèle sera appliqué à toutes les fenêtres qui l'utilisent. Son rôle principal est de respecter la charte graphique d'une application.	WDT

CHAPITRE 2

Développement mobile multiplateforme

Composant externe	Brique logicielle composé d'un ensemble d'éléments Windev mobile, redistribué pour d'autres développeurs afin d'intégrer une fonctionnalité.	WDI
Composant interne	Regroupement d'éléments d'un projet. Il permet d'organiser le projet par fonctionnalités et il peut être intégré dans d'autres projets. Sa différence avec un composant externe est qu'il peut être débogué directement.	WCI
Modèle UML	Diagramme UML associé à la conception.	UML

Tableau 2 : Tableau récapitulatif des éléments d'un projet Windev mobile

6. Architecture d'une application Windev Mobile

Le développement sur Windev mobile suit une architecture 3 tiers, ainsi une application est composée de trois couches :

- **La couche présentation** : Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. Autrement l'interface homme machine.
- **La couche traitements** : Elle correspond à la partie fonctionnelle de l'application. Autrement l'application opère sur les données en fonction des actions des utilisateurs, effectuées au travers de la couche présentation
- **La couche accès aux données** : correspond aux données destinées à être enregistrées de manière permanente, sur des fichiers ou des bases de données [11].

CHAPITRE 2

Développement mobile multiplateforme

Les trois couches communiquent entre elles à l'aide de fonctions spécifiques et avancées du WLangage.

Les avantages de cette séparation sont :

- L'indépendance entre les données et les traitements.
- Une sécurité plus importante car l'accès à la base de données n'est autorisé que par la couche Traitements.
- L'optimisation le travail en équipe et le développement multi-cibles.

La figure 8 illustre l'architecture 3 tiers d'une application Windev mobile.

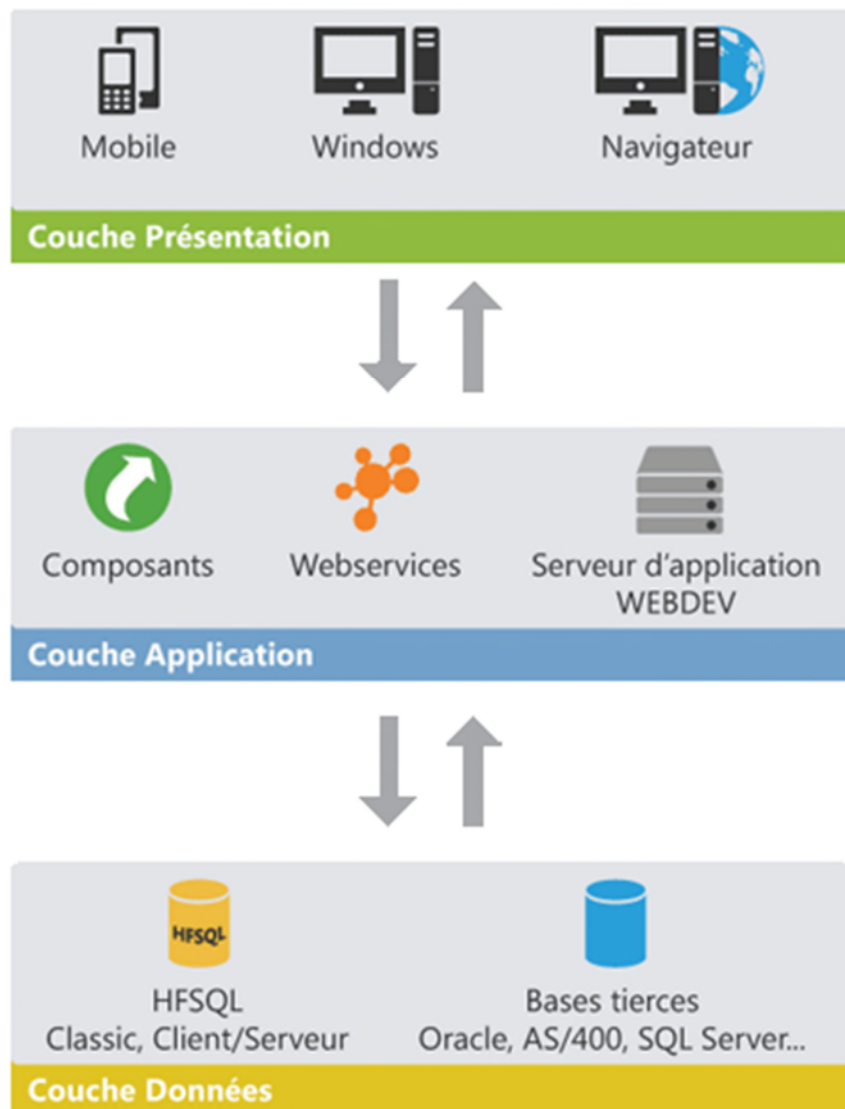


Figure 8 : Architecture d'une application mobile

7. WLangage

7.1. Définition

Le WLangage est un langage de programmation de 5^{ème} génération indus dans les outils de développement WinDev Mobile. Il est propriétaire et ne peut être manipulé qu'avec les outils PC SOFT. Le WLangage est né en 1992 avec la première version de Windev.

Le WLangage peut également s'appuyer sur le Framework Java pour une partie de ses fonctionnalités. Ce qui permet une indépendance relative et limitée du fichier exécutable par rapport au système d'exploitation cible.

Le WLangage comporte plus de :

- 2 800 fonctions
- 3 500 constantes
- 450 variables
- 490 propriétés
- 160 types de variables avancés proposant un ensemble de propriétés
- 120 mots-clés. [12]

7.2. Les variables

Windev propose des différents types de variables (booléen, entier, monétaire, chaîne, date, heure, durée, dateheure, durée, variant, tableau, structure, ...)

La syntaxe à utiliser pour déclarer un variable est très simple : il suffit d'indiquer le nom de la variable et son type :

Exemple:

NomClient est une chaîne

Compteur est un entier

i, j, k sont des entiers

7.3. Les constantes

Les constantes sont des éléments du langage dont la valeur est fixée une fois pour toutes.

7.4. Les instructions de base

Le WLangage permet de gérer les instructions conditionnelles du type :

- **SI, SINON, FIN** pour effectuer un test sur une condition
- **SELON, CAS, FIN** pour exécuter une ou plusieurs actions selon les différents résultats d'un test sur une condition.
- **Les instructions de boucle :**

Le WLangage propose plusieurs possibilités pour gérer les boucles

- **POUR, FIN** pour un nombre d'itération déterminé.
- **TANTQUE, FIN** pour un nombre d'itération indéterminé et dont la condition de la sortie est testée au début de la boucle.
- **BOUCLE, FIN** pour un nombre d'itération indéterminé et dont la condition de la sortie est testée dans la boucle. L'instruction **SORTIR** permet de sortir de cette boucle.

Le WLangage propose également des boucles de type **POUR TOUT, FIN** permettant de parcourir les éléments d'un champ, les chaînes de caractères, les enregistrements des fichiers. [12]

7.5. Les fonctions

Côté programmation, il existe les procédures et les fonctions :

Les fonctions renvoient un résultat et Les procédures servent à exécuter un traitement spécifique.

Windev permet de gérer simplement les deux types de traitements, de manière identique : Sous Windev il n'y a pas de différence entre fonction et procédure.

7.6. Les modes de programmation

Deux méthodes de programmation sont possibles dans WinDev :

- Programmation procédurale
- Programmation Orientée Objet (POO)

Les deux méthodes peuvent se mixer dans un même programme développé avec WinDev [12].

Partie 3 : Bases de données dans Windev mobile

1. Généralités

Les bases de données sont une brique essentielle sur Windev mobile. Ce dernier permet l'accès et l'utilisation de la totalité des bases de données existantes sur le marché, mais également il offre aux développeurs une puissante base de données HFSQL livrée en standard avec l'environnement.

2. Structure d'une base de données

La création d'une base de données sous Windev Mobile correspond à la création d'une analyse. Ainsi un projet Windev mobile est lié à une analyse.

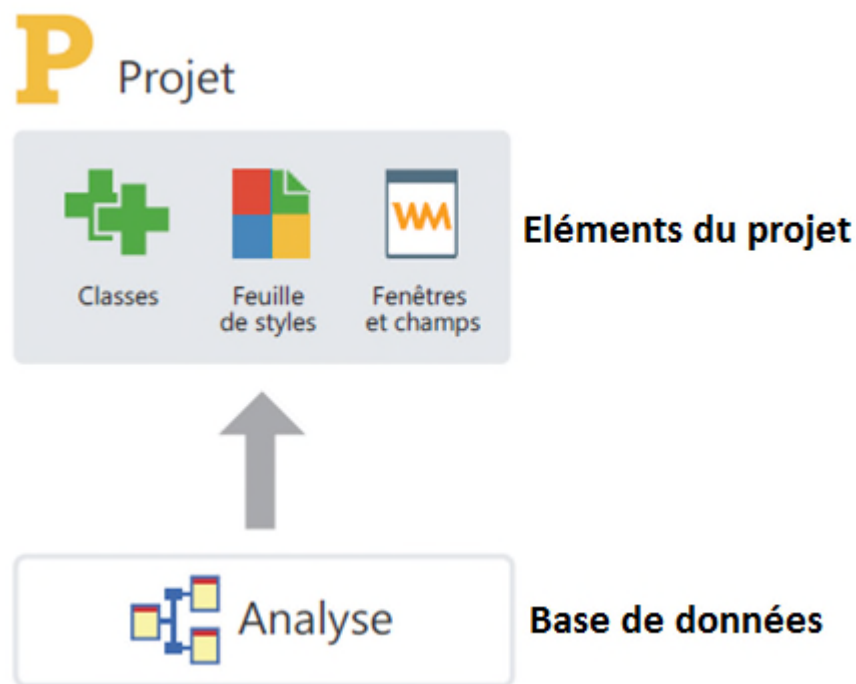


Figure 9 : Structure d'un projet Windev mobile

Windev mobile utilise un vocabulaire différent pour décrire les concepts liés aux bases de données. Et le tableau 3 illustre ces différents concepts.

CHAPITRE 2

Développement mobile multiplateforme

Concept	Vocabulaire de Windev mobile
Base de données	Analyse
Graphe de l'analyse	Modèle du schéma
Table	Fichier de données
champ	Rubrique
colonne	Rubrique clé
Plusieurs colonnes	Clé composée
Ligne	Enregistrement
Relation	Liaison
Clé primaire	Clé unique
Clé étrangère	Rubrique de liaison
Clé étrangère sans contrainte d'unicité	Clé avec doublon
Index	Clé

Tableau 3 : Vocabulaire de Windev mobile lié aux concepts de bases de données

Une analyse est un ensemble de fichiers de données reliés par des liaisons. Un fichier de données est un ensemble de données structurées en rubriques et composé d'enregistrements.

Le croisement d'une rubrique clé et d'un enregistrement est une valeur de rubrique.

Une clé est un moyen d'accélérer les recherches, les requêtes et les accès à un fichier de données. Une clé peut être définie sur une rubrique clé ou sur une clé composée. Une clé unique permet d'identifier un fichier de données. Une rubrique de liaison est utilisée conjointement à une clé unique pour réaliser une liaison entre deux fichiers de données[12].

La figure 10 illustre les différents concepts d'une base de données sous Windev mobile.

CHAPITRE 2

Développement mobile multiplateforme

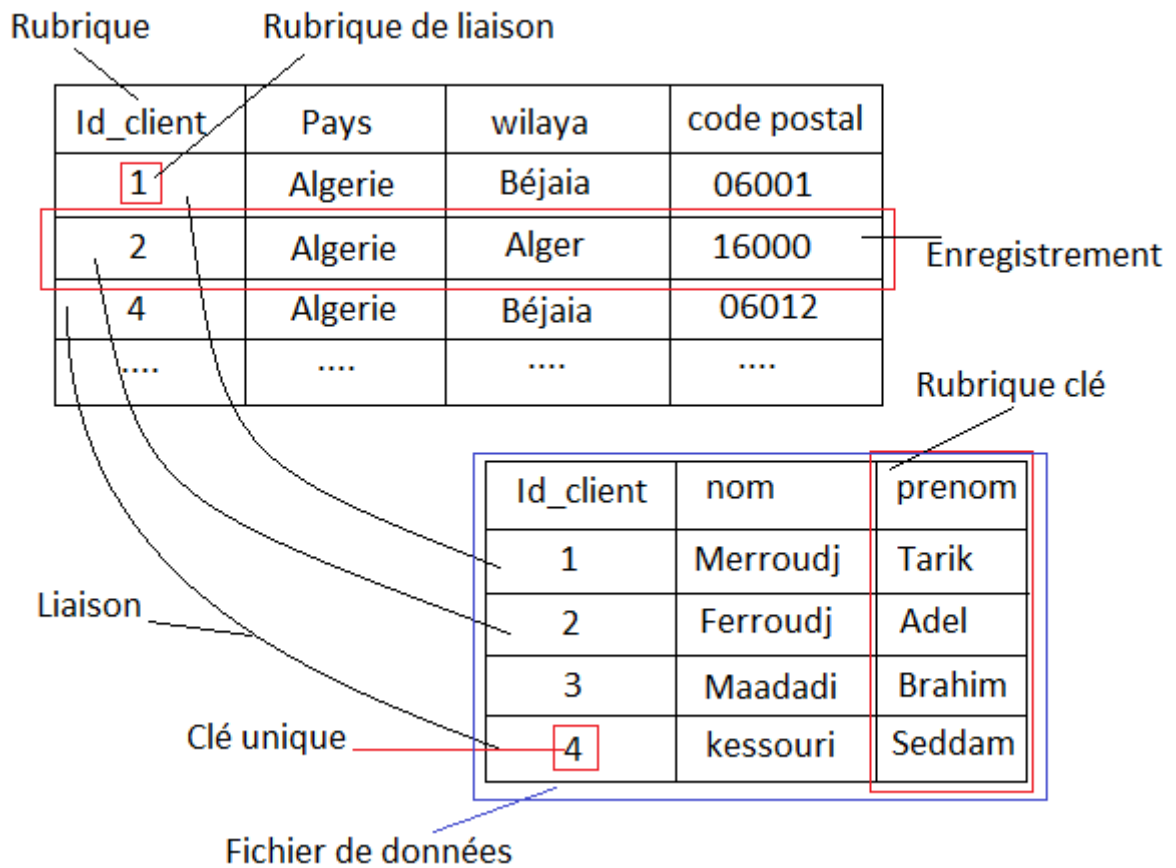


Figure 10 : Exemple d'une base de données

3. Cycle de création d'une base de données

La création d'une base de données consiste à créer une analyse qui va contenir toutes les structures de données. Windev mobile fournit un outil très puissant : l'éditeur d'analyse, qui permet de créer l'analyse et de manipuler toutes les structures de données.

Pour décrire une analyse, il existe trois démarches possibles :

- Première démarche : Consiste à créer l'analyse directement, en créant le modèle logique des données(MLD).
- Deuxième démarche : Consiste à créer en premier lieu le modèle conceptuel des données (MCD) et ensuite générer automatiquement le modèle logique des données (MLD).

CHAPITRE 2

Développement mobile multiplateforme

- Troisième démarche : Consiste à implémenter la base de données en utilisant un outil externe à Windev mobile puis importer le fichier de description, comme le montre la figure 11.

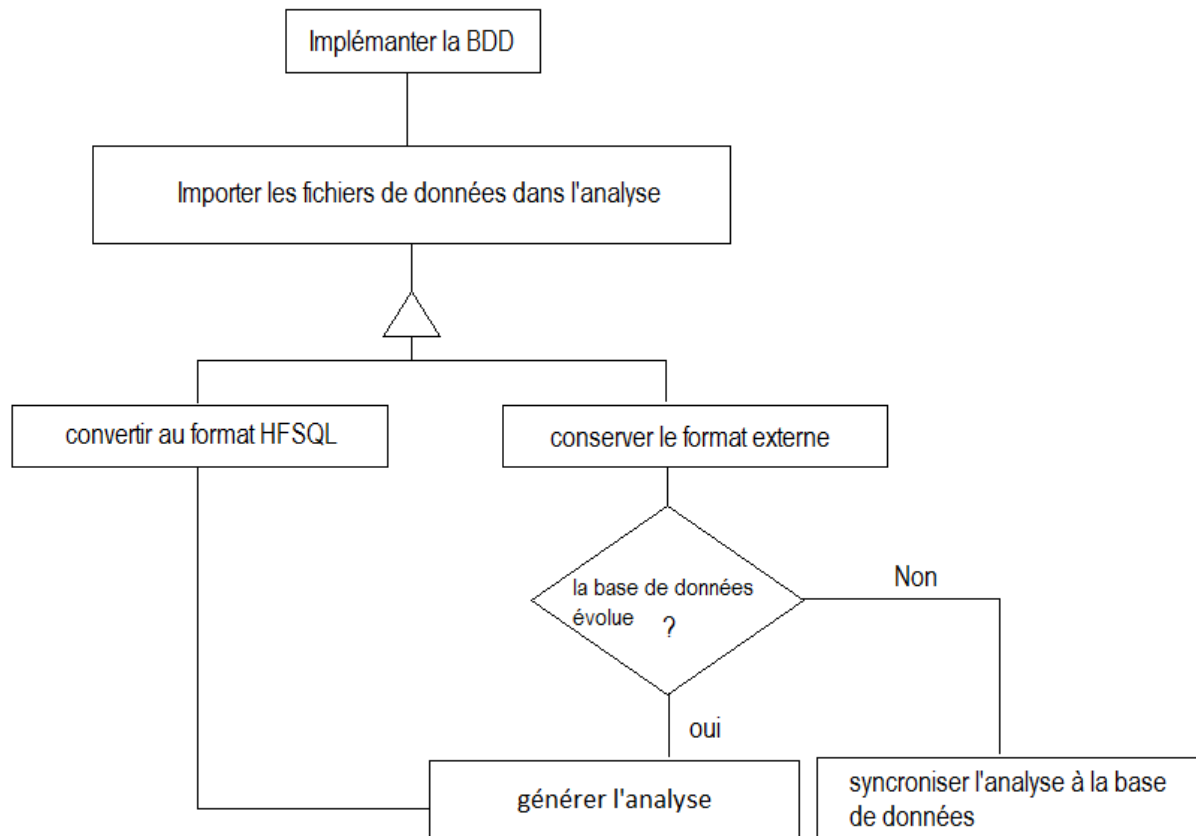


Figure 11 : importation d'une base de données

4. Bases de données tierces sous Windev mobile

Windev mobile propose trois modes d'accès à toutes les bases de données tierces :

- **Accès natif** : C'est le mode d'accès le plus rapide. Pour chaque type d'accès natif, il est nécessaire de posséder un module complémentaire à Windev mobile.
- **Accès par un driver ODBC direct** : Les caractéristiques de la connexion à la base de données doivent être définies dans l'administrateur ODBC de Windows. Seules les fonctions de type SQL sont utilisables pour ce type d'accès.

CHAPITRE 2

Développement mobile multiplateforme

- **Accès par un provider OLE DB** : Ce type d'accès utilise un provider OLE DB. Il est nécessaire d'installer le provider ainsi que le MDAC pour utiliser cet accès.

Windev mobile permet de manipuler les bases de données en utilisant les fonctions SQL ou HFSQL sauf pour celles qui utilisent un accès ODBC, elles ne sont manipulés qu'en utilisant les fonctions SQL [12].

Le tableau 4 illustre de manière non exhaustive quelques fonctions pour la manipulation des bases de données.

	Fonctions SQL	Fonctions HFSQL
Connexion à la base de données	- SQLConnecte	- HOuvreConnexion
Exécution de la requête	- SQLExec	- HExécuteRequête - HExécuteRequêteSQL
Fermeture de la connexion	- SQLDéconnecte	- HFermeConnexion

Tableau 4 : Fonctions essentielles pour manipuler les bases de données tierces

La figure 12 illustre les différentes bases de données supportées par Windev mobile et les modes d'accès.

CHAPITRE 2

Développement mobile multiplateforme

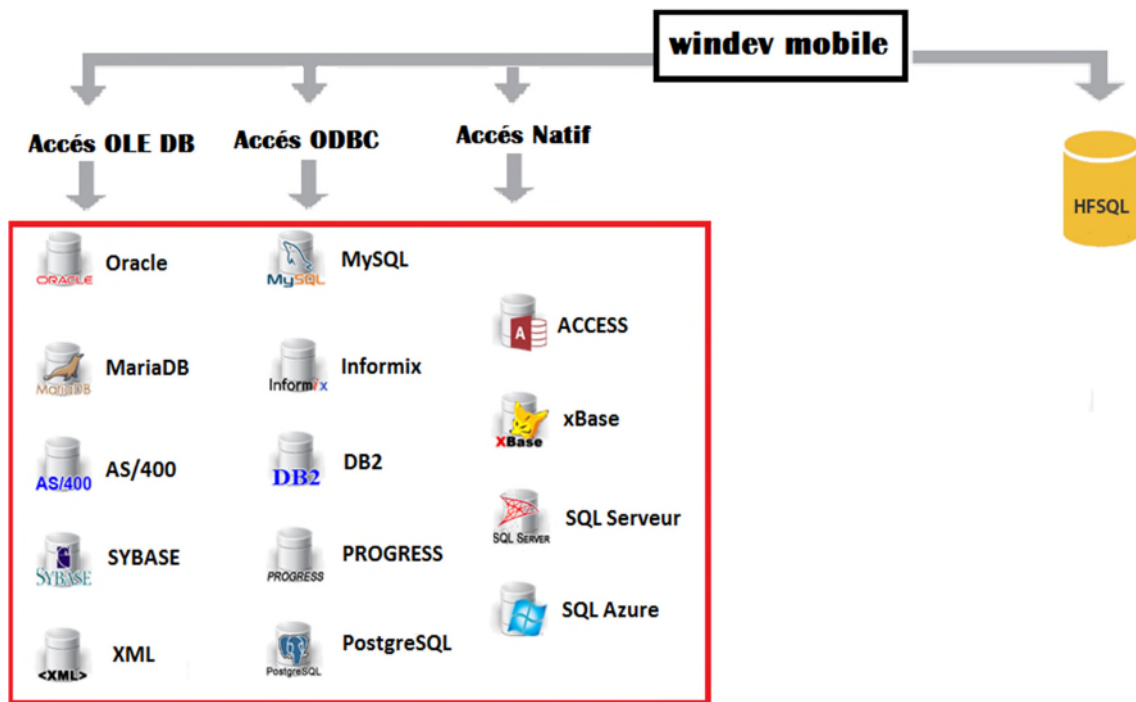


Figure 12 : Bases de données accessibles par Windev mobile et modes d'accès

5. Base de données HFSQL

5.1. Présentation

HyperFileSQL est le moteur de base de données incorporé en standard dans l'environnement WinDev mobile, édité en 1988 par la société française PC SOFT dans l'objectif d'être utilisé dans les applications créées avec WinDev mobile.

C'est un puissant SGBDR (Système de Gestion de Base de Données Relationnelle), adapté à tous les types d'applications: applications métiers, applications critiques temps réel, progiciels, serveurs d'applications, serveurs Web, périphériques mobiles.

5.2. Les avantages de HFSQL

HFSQL est un choix éprouvé pour les bases de données et largement adapté pour le développement sous Windev mobile. Il présente un grand nombre d'avantages, on cite en autre autres :

- La richesse fonctionnelle : Gestion des intégrités référentielles, journalisation, réplication, Gestion des transactions et des triggers et cryptage des données.
- Gratuité, livraison en version complète, illimitée, avec tous les outils et intégration totale avec Windev mobile.
- Inclusion d'un ensemble d'outils utiles et livrés en standard : import de données, vérification de base, visionnement des données, description de schémas des données et optimisation de performances.
- Compatibilité avec toutes les plateformes et entre les différentes versions : locale, réseau, embarquée, client/serveur, cluster.
- Reprise immédiate et automatique en cas de déconnexion et insensibilité aux arrêts intempestifs des postes clients.
- Sécurité, cryptage des communications et protection contre les injections SQL.
- Installation, administration et maintenance automatique.
- Optimisation des requêtes et des temps de réponse.
- Capacité de stockage immense.
- Gestion efficace des langues et différents jeux de caractères ainsi que des ordres de tri.
- Faible consommation des ressources.
- Facilité de déploiement et de prise en main.
- Support technique et documentation disponible.
- Multi -bases et multi-serveurs : une même machine peut héberger plusieurs serveurs HFSQL et chaque serveur peut contenir plusieurs bases de données[13].

5.3. Les caractéristiques techniques de HFSQL

HFSQL offre des capacités de stockage élevées et un moteur de bases de données très puissant. Il fonctionne sur toutes les plateformes et sur tous les réseaux et gère automatiquement plusieurs centaines d'accès simultanés.

Le tableau 5 illustre les caractéristiques techniques essentielles de HFSQL.

Le moteur SQL	<ul style="list-style-type: none"> - Il utilise de puissants mécanismes d'optimisation de recherche : les statistiques des index et les contraintes de choix d'index d'accès les plus discriminants lors des requêtes. - Utilise le mécanisme de gestion avancée des caches mémoire pour améliorer les performances. - Il effectue automatiquement la gestion et la répartition de la charge: si un client exécute de très nombreuses requêtes consommatrices de ressources, le serveur équilibre automatiquement la charge.
Langage SQL	<ul style="list-style-type: none"> - La norme ANSI SQL 92. - support complet de nombreuses syntaxes supplémentaires et spécifiques de SQL Server et Oracle. - Gestion des sous requêtes et des requêtes imbriquées - Gestion des opérateurs ensemblistes: union, cartésien, jointure, jointure externe. - Gestion des opérateurs d'agrégation: count, sum, avg, min, max, écart-type, variance et des opérateurs de tri et de regroupement
Capacités de stockage	<ul style="list-style-type: none"> - Le nombre de colonnes par table : 65535 colonnes. - Le nombre d'index par table : 65535 index. - Le nombre de lignes par table : 329 millions de milliards lignes. - La taille d'une table : 4 millions de téras. - Taille d'une ligne : 2 Go. - Taille d'une colonne : 65535 octets.

Tableau 5 : Caractéristiques techniques de HFSQL

5.4. Les modes de fonctionnement d'une application HFSQL

Une application mobile peut fonctionner en deux modes : Le mode client/serveur ou le mode embarqué, autrement appelé Classic.

Une application en mode Classic ne nécessite aucun accès à un serveur distant et utilise une base de données HFSQL Classic.

Par contre une application HFSQL Client/serveur consiste à exécuter l'application sur différents terminaux mobiles (clients) et à déporter la ou les bases de données et les traitements sur un serveur distant. Ce mode de fonctionnement permet des temps de réponses plus rapides et plus fiables, ainsi qu'une maintenance de la base de données facilitée.

5.4.1. HFSQL Classic

Le format HFSQL Classic (également appelé HFSQL Mobile) est le format de base de données fourni avec WINDEV Mobile. Il s'agit d'un SGBD Relationnel redistribuable gratuitement et compatible avec toutes les plateformes mobiles : Windows Mobile, iPhone, iPad, Android et Universal Windows 10 App.

Cependant, la taille disponible sur les périphériques mobiles étant souvent restreinte, les fonctionnalités suivantes ne sont pas gérées par HFSQL Classic sur mobile :

- les transactions.
- la journalisation.
- la gestion des blocages des fichiers et des enregistrements.
- la gestion des fichiers au format Hyper File 5.5.

Une base de données HFSQL Classic correspond à un ensemble de fichiers ".FIC", ".NDX", ".MMO", ".NTX".

Chaque fichier de données peut être manipulé par une application WINDEV Mobile. Ces manipulations sont réalisées grâce aux fonctions HFSQL comme `HOuvreConnexion`, `HExécuteRequête`, `HExécuteRequêteSQL` [12].

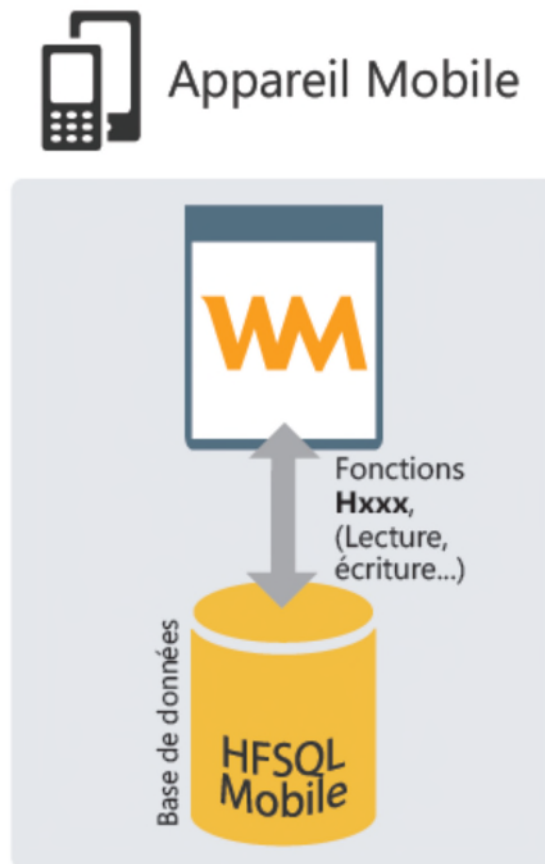


Figure 13 : Manipulation d'une base de données HFSQL mobile

5.4.2. HFSQL client/serveur

Une application HyperFileSQL Client/serveur consiste à établir une communication entre l'application cliente installée sur le terminal mobile et le serveur HFSQL qui contient toutes les tables de la base de données.

Seul le serveur qui accède physiquement aux tables de données, autrement dit toutes les manipulations sur la Base de données et toutes les requêtes sont exécutées sur le serveur.

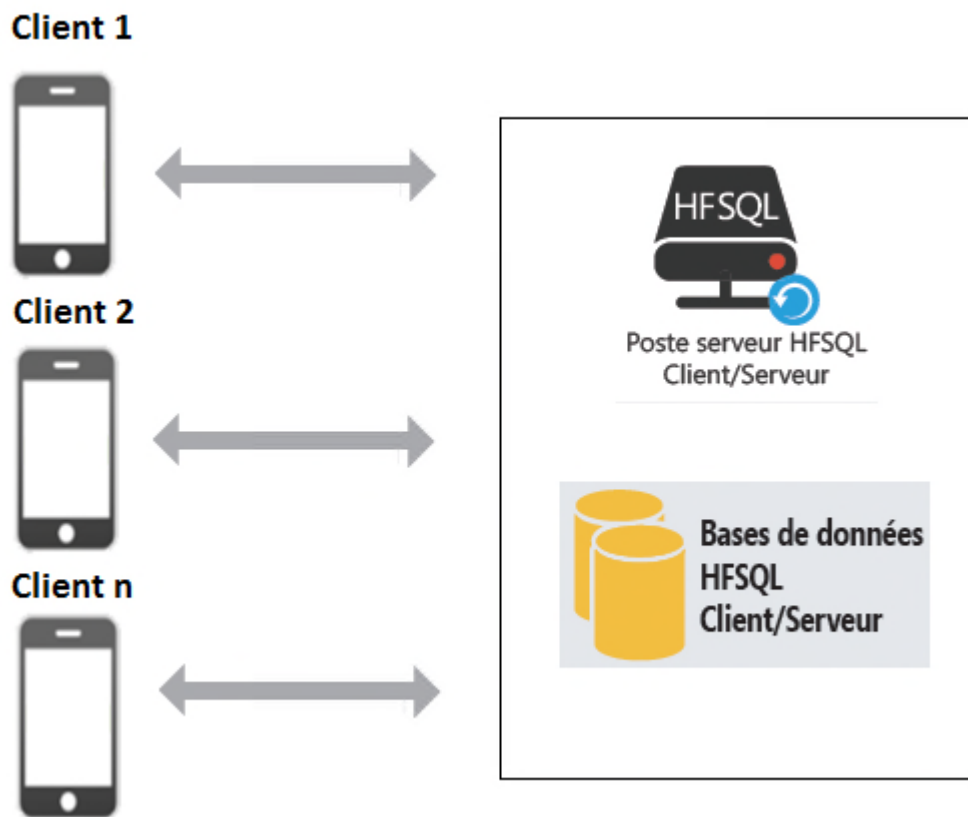


Figure 14 : Fonctionnement d'une application client/serveur

5.5. Outils de HFSQL

HFSQL comporte plusieurs outils, parmi ces outils essentiels :

- Administration : Centre de Contrôle HFSQL

Le Centre de Contrôle HFSQL est un outil d'administration indispensable disposant d'une interface graphique intuitive et ergonomique. Le Centre de Contrôle HFSQL permet d'effectuer de nombreuses tâches.

- WDMAP: visualiseur de données

L'outil WDMAP permet de visualiser, éditer et modifier les données d'une table. WDMAP est très utile en phase de test et de débogage. WDMAP

CHAPITRE 2

Développement mobile multiplateforme

permet de filtrer, de trier les données, d'effectuer des exports immédiats (vers Word, Excel, OpenOffice, XML).

- Outil de comparaison de données : WDHFDiff

L'utilitaire WDHFDIFF permet de comparer :

- la structure de 2 tables,
- les données de 2 tables

- Robot de surveillance

Permet de sécuriser l'exploitation des serveurs. Le robot surveille en permanence, et détecte immédiatement les éventuelles connexions impossibles avec le serveur.

- SILO : superviseur d'infrastructure logicielle

Le SILO permet de recenser les accès de toutes les applications WINDEV Mobile dont les bases de données HFSQL. Le SILO permet d'analyser les accès aux bases de données au cours du temps, de savoir quelles applications et quels postes accèdent à une base de données.

5.6. Fonctions HFSQL

le Wlangage offre un large choix de fonctions pour manipuler les bases de données HFSQL, la convention de nommage est qu'elle commence par la lettre "H".

le tableau suivant illustre les fonctions essentielles de HFSQL

Fonctions	utilisation
HAjoute	Ajoute l'enregistrement présent en mémoire dans le fichier de données (la requête ou la vue).
HExécuteRequêteSQL	Initialise une requête écrite en langage SQL et déclare cette requête au moteur HFSQL.

CHAPITRE 2

Développement mobile multiplateforme

HLitRecherchePremier	Positionne sur le premier enregistrement du fichier dont la valeur d'une rubrique spécifique est strictement égale à une valeur recherchée (recherche à l'identique).
HModifie	Modifie l'enregistrement spécifié ou l'enregistrement présent en mémoire dans le fichier de données (la requête ou la vue).
HNumEnr	Renvoie le nombre d'enregistrements d'un fichier, d'une requête ou d'une vue HFSQL : enregistrements actifs, rayés, supprimés, etc.
HOuvre	Ouvre un fichier de données.
HOuvreConnexion	Ouvre une connexion à une base de données spécifique.
HRecherchePremier	Positionne sur le premier enregistrement du fichier dont la valeur d'une rubrique spécifique est supérieure ou égale à une valeur recherchée
HSupprime	Supprime un enregistrement d'un fichier de données (d'une requête ou d'une vue).
HTrouve	Vérifie si l'enregistrement en cours correspond au filtre ou à la recherche en cours.

Tableau 6 :les fonctions essentielles de HFSQL

5.7. Centre de contrôle HFSQL

Le centre de contrôle HFSQL est un outil redistribuable, il peut être lancé depuis Windev mobile ou en lançant directement le programme "CCxxOHF.EXE".

Il permet de manipuler :

- Le serveur : Arrêt, redémarrage, configuration du serveur, définition des utilisateurs, tâches planifiées et sauvegardes.

- Les bases de données : Descriptions, recherche, sauvegarde, maintenance des fichiers de données et transactions.
- Les fichiers de bases de données : Description, contenu, recherche, journal et transactions [12].

6. Les requêtes

Une requête sert à interroger une base de données pour visualiser, insérer, modifier ou supprimer des données. La structure de la requête définit les données manipulées.

Une requête peut interroger un ou plusieurs fichiers de données.

L'éditeur de requêtes permet de créer très simplement des requêtes, sans avoir à programmer.

6.1. Types de requêtes HFSQL

L'éditeur de requêtes permet de créer plusieurs types de requêtes :

-requête de sélection : ce type de requêtes permet de visualiser une sélection d'enregistrements provenant d'un ou de plusieurs fichiers d'une base de données. C'est le type de requête le plus utilisé.

Exemple: nous voulons obtenir la somme totale des commandes par client habitant Londres. Seuls les clients ayant un cumul des commandes supérieur à 3000 Euros seront conservés. Le résultat sera trié par ordre croissant du cumul des commandes. Pour réaliser cette requête, le résultat sera automatiquement regroupé par nom du client et par ville.

CHAPITRE 2

Développement mobile multiplateforme

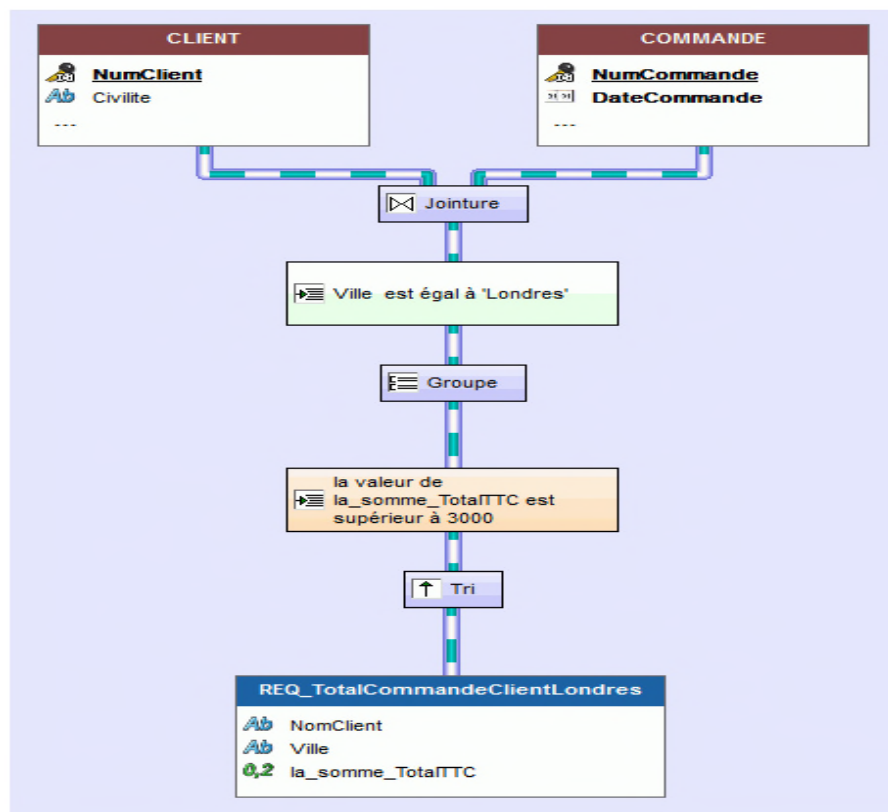


Figure 15 : Exemple d'une requête de SELECTION

- le code SQL généré pour la requête "sélection " est le suivant:

```
SELECT
  CLIENT.NumClient AS NomClient,
  CLIENT.VILLE AS VILLE,
  SUM(COMMANDE.TotalTTC) AS la_somme_TotalTTC
FROM
  CLIENT,
  COMMANDE
WHERE
  CLIENT.NumClient = COMMANDE.NumClient
  AND
  (
    CLIENT.VILLE = 'Londres'
  )
GROUP BY
  CLIENT.NumClient,
  CLIENT.VILLE
HAVING
  SUM(COMMANDE.TotalTTC) > 3000
ORDER BY
  la_somme_TotalTTC ASC
```

Figure 16 : code SQL de la requête SELECTION

-requête d'insertion : ce type de requête permet d'ajouter des enregistrements dans un fichier d'une base de données.

CHAPITRE 2

Développement mobile multiplateforme

Cette figure montre comment insérer un nouveau client dans le fichier "client".

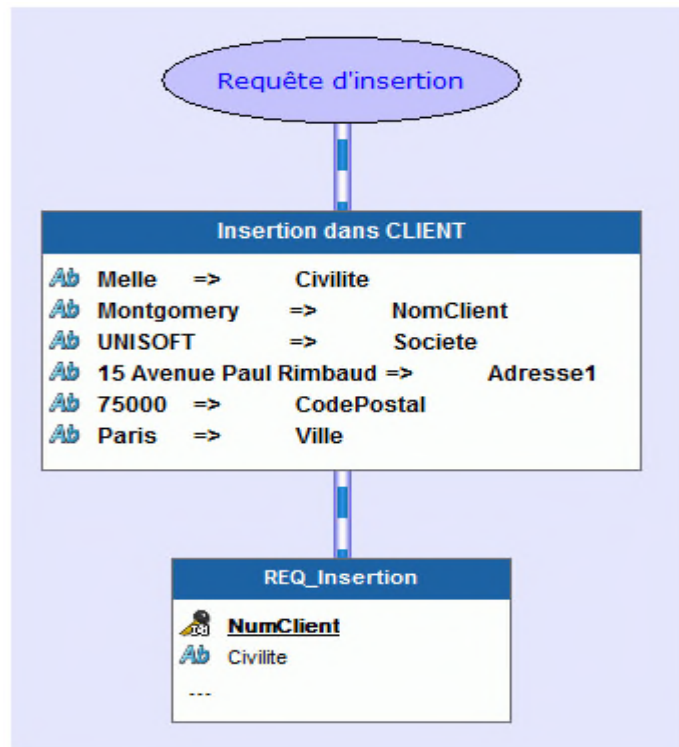


Figure 17 : Exemple d'une requête INSERTION

- le code SQL généré pour la requête "insertion " est le suivant:

```
INSERT INTO CLIENT
(
    Civilete,
    NomClient,
    Societe,
    Adresse1,
    CodePostal,
    Ville
)
VALUES
(
    'Melle',
    'Montgomery',
    'UNISOFT',
    '15 Avenue Paul Rimbaud',
    '75000',
    'Paris'
)
```

Figure 18 : code SQL de la requête INSERTION

-requête de modification : ce type de requête permet de modifier des enregistrements présents dans un fichier d'une base de données.

CHAPITRE 2

Développement mobile multiplateforme

Pour obtenir les modifications de la figure 19, les principales caractéristiques de la requête sont les suivantes :

- le fichier dans lequel des données vont être modifiées.
- les rubriques dans lesquelles des valeurs vont être modifiées.
- les conditions de sélection des enregistrements (clients habitant Nantes, produits dont le prix est supérieur à 40 Euros).

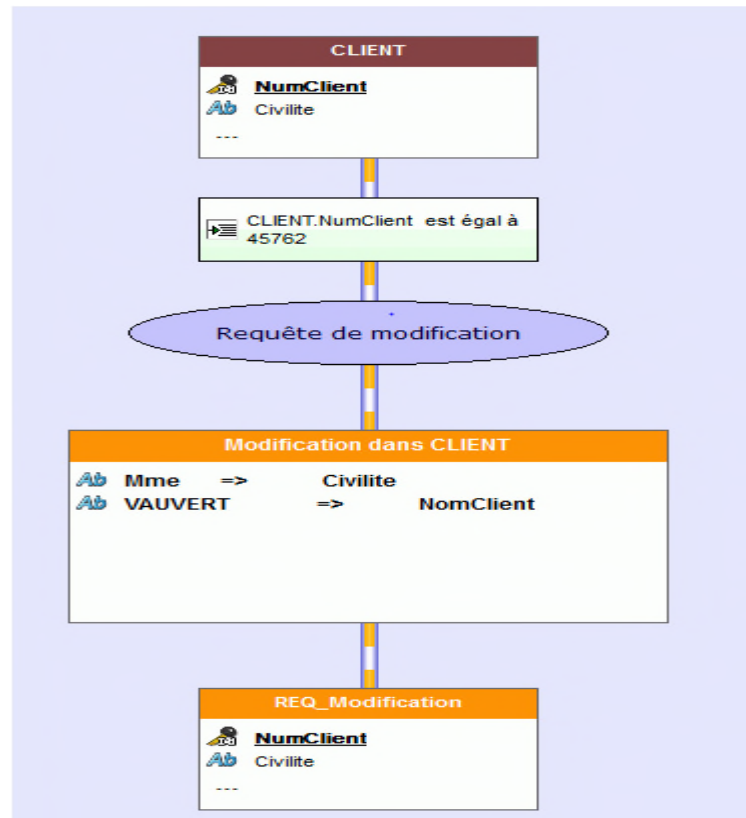


Figure 19 : Exemple d'une requête MODIFICATION

- le code SQL généré pour la requête "modification " est le suivant:

```
UPDATE
  CLIENT
SET
  Civilite = 'Mme',
  NomClient = 'VAUVERT'
WHERE
  CLIENT.NumClient = 45762
```

Figure 20 : code SQL de la requête MODIFICATION

-requête de suppression : ce type de requêtes permet de supprimer des enregistrements présents dans un fichier d'une base de données.

CHAPITRE 2

Développement mobile multiplateforme

Pour effectuer la suppression de la figure 21, les principales caractéristiques de la requête de suppression sont les suivantes :

- le fichier dans lequel des données vont être supprimées.
- les conditions de sélection des enregistrements à supprimer (clients habitant Nantes, produits dont le prix est supérieur à 40 Euros).

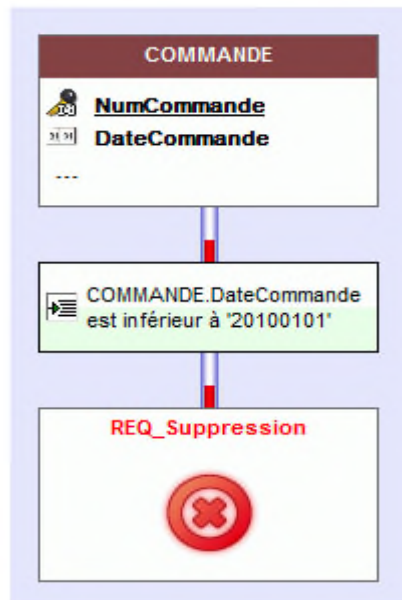


Figure 21 : Exemple d'une requête SUPPRESSION

- le code SQL généré pour la requête "suppression " est:

```
DELETE FROM  
  COMMANDE  
WHERE  
  COMMANDE.DateCommande < '20100101'
```

Figure 22 : code SQL de la requête SUPPRESSION

-requête composée : ce type de requêtes permet de combiner des requêtes de sélection.

Exemple: nous voulons sélectionner des clients dont le nom commence par la lettre 'A' et des prospects habitant Nice. Pour réaliser cette sélection, il est nécessaire de combiner les deux requêtes suivantes :

CHAPITRE 2

Développement mobile multiplateforme

- sélection des clients dont le nom commence par la lettre 'A'. Cette requête est basée sur le fichier "Client".
- sélection des prospects habitant Londres. Cette requête est basée sur le fichier "Prospect".

Les fichiers "Client" et "Prospect" sont de même structure dans l'analyse.

Les enregistrements communs aux deux fichiers ("Client" et "Prospect") peuvent :

- soit apparaître une seule fois dans le résultat de la requête (opérateur UNION).
- soit apparaître deux fois dans le résultat de la requête (opérateur UNION ALL).

Pour combiner des requêtes, il est nécessaire de connaître :

- le nom des requêtes à combiner.
- si les enregistrements communs doivent apparaître une seule ou plusieurs fois dans le résultat de la requête

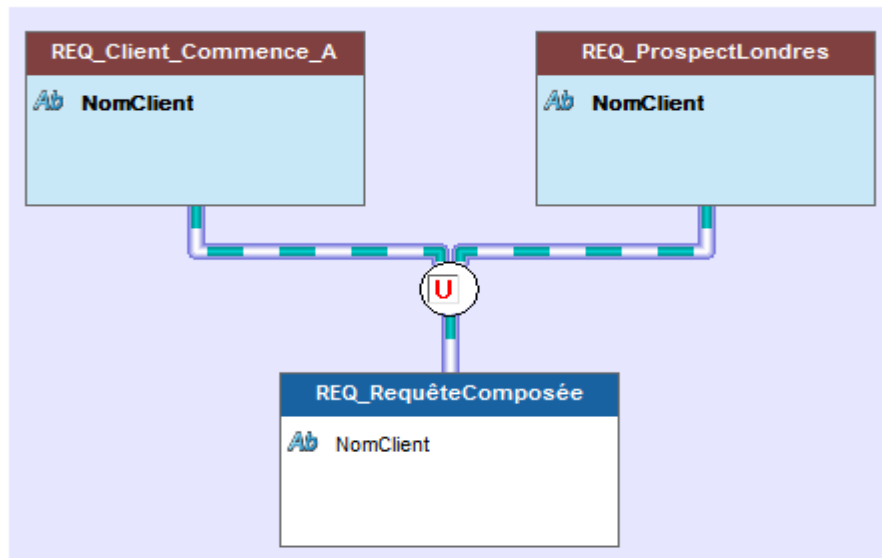


Figure 23 : Exemple d'une requête composée

- le code SQL généré pour la requête " composée " est le suivant:

```
SELECT
    CLIENT.NomClient AS NomClient
FROM
    CLIENT
WHERE
    CLIENT.NomClient LIKE 'A%'

UNION

SELECT
    PROSPECT.NomClient AS NomClient
FROM
    PROSPECT
WHERE
    PROSPECT.VILLE = 'Londres'
```

Figure 24 : code SQL de la requête composée

6.2. La création de requêtes

Pour créer une requête, deux méthodes sont possibles :

- création de la requête avec un assistant ou à l'aide de la fenêtre de description, et génération automatique du code SQL.
- saisie du code SQL.

4.3. L'éditeur de requêtes

L'éditeur de requêtes permet de créer automatiquement des requêtes sur les fichiers de données. La création d'une requête est très simple et intuitive. Elle est automatiquement générée (en code SQL optimisé) et peut être testée immédiatement.

La programmation est alors simplifiée : fenêtres, pages, tables, combos, états du projet pourront être basés sur les requêtes.

Conclusion

Dans ce chapitre nous avons recensé et présenté quelques aspects essentiels de l'environnement de développement Windev Mobile et les différents outils qu'il offre pour la conception et la gestion des bases de données.

Les chapitres suivants sont consacrés à l'étude de cas relative au modélisation et conception d'une base de données pour une application mobile pour la mise en relation d'affaires suivant l'AGL Windev Mobile.

CHAPITRE III :
étude de cas- modélisation et
conception de la base de données
de l'application yperesia

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

Introduction

Dans ce chapitre nous présentons la méthodologie de travail qu'on a suivi pour réaliser notre projet qui consiste à la modélisation et la conception de la base de données pour une application mobile pour la mise en relation d'affaires sous Windev, que nous déploierons sur la plateforme Android.

1. Cadre de notre projet

Nous avons réalisé notre travail dans le cadre du projet de fin d'études en informatique spécialité génie logiciel.

Notre travail est basé sur l'application des aspects essentiels de l'ingénierie du logiciel :

- l'application d'un processus de développement adapté à notre projet.
- l'utilisation d'un atelier de génie logiciel.
- Travail en équipe et communication.

Nous avons choisi de répartir notre projet entre deux binômes :

- Binôme 1 : Analyse des besoins, prototypage et maquettage des IHM.
- Binôme 2 : Modélisation et conception des bases de données.

Nous avons réparti les taches comme suit :

Tache	intervenants
Spécification des besoins et planification	Binôme 1, binôme 2
Analyse des besoins, prototypage et maquettage des IHM	Binôme 1
Modélisation et conception des bases de données	Binôme 2
Implémentation et tests	Binôme 1, Binôme 2

Tableau 7 : répartition des taches

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

2. Démarche adoptée

Nous avons adopté une démarche itérative et incrémentale.

Nous avons établi un tableau d'exigences fonctionnelles, que nous avons classé selon les priorités et les risques.

Ensuite nous avons planifié les différentes itérations (versions) et on a opté pour les activités nécessaires.



Figure 25 : Démarche adoptée

Nous avons planifié trois itérations :

- première itération : consiste à développer le noyau (version 1) qui contient les fonctionnalités essentielles.
- Deuxième Itération : consiste à développer la version 2.
- Troisième itération : consiste à développer la version finale qui contient toutes les fonctionnalités.

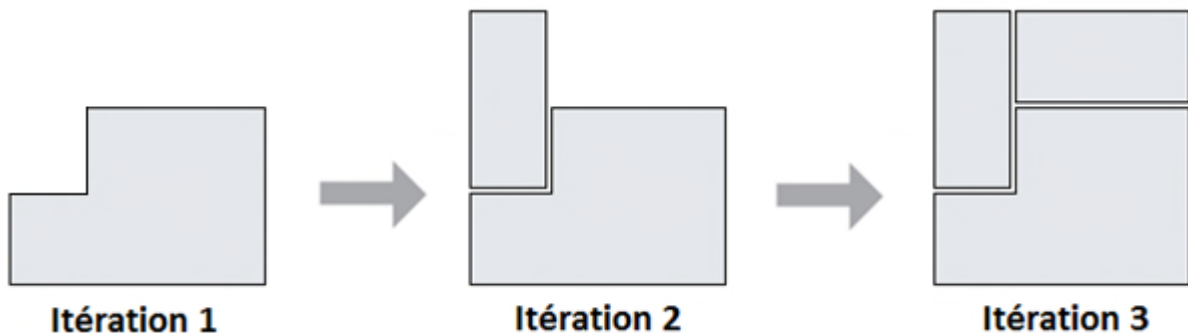


Figure 26 : Les différentes versions de notre application

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

3. Description du contexte de l'application

Un citoyen a besoin de différents services et a tout moment : un médecin, une pharmacie, un hôtel, un enseignant, un artisan...etc. Il a besoin de trouver le fournisseur du service le plus proche, celui qui correspond mieux à ses attentes et le plus rapidement possible.

Le fournisseur de service qui peut être un artisan, un médecin, un enseignant, une entreprise, un hôtel, etc....veut promouvoir son activité et trouver de la clientèle et pour cela il doit présenter son service et aider le client à le trouver et le contacter.

Le travail à réaliser consiste à modéliser, concevoir et implémenter la base de données de l'application mobile "ypiresia" pour la mise en relation du consommateur et du fournisseur du service.

La base de données doit permettre à l'application ypiresia d'enregistrer les clients et les fournisseurs de service et doit fournir aux clients les résultats de recherches appropriées.

4. Acteurs de la base de données

Nous avons identifié trois acteurs externes qui vont interagir avec la base de données : Le fournisseur de service, Le client et le visiteur. La figure 27 illustre ces trois acteurs.

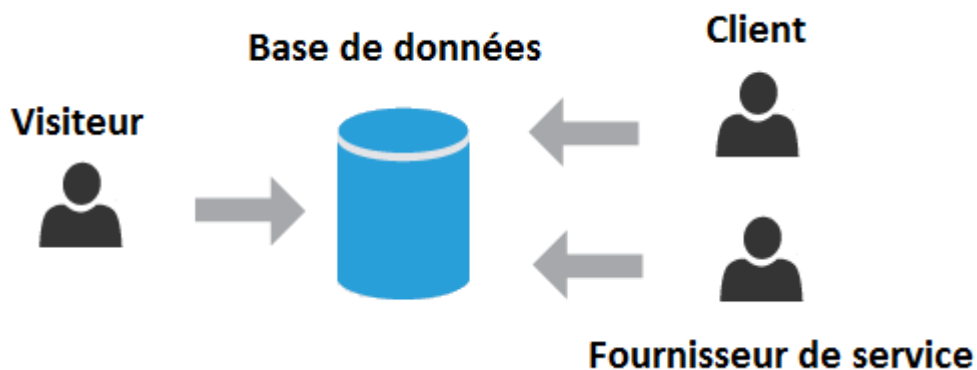


Figure 27 : Les acteurs qui interagissent avec la base de données

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

Le tableau 8 donne une vision globale des différentes actions des acteurs avec la base de données.

Acteur	Actions
Fournisseur de service	<ul style="list-style-type: none">- Chaque fournisseur crée un enregistrement sur la base de données lors de l'inscription, il peut générer des requêtes de mise à jour, et il peut proposer des offres aux clients.
Visiteur	<ul style="list-style-type: none">- Génère des requêtes de sélection simples pour effectuer des recherches simples et basiques sur les fournisseurs de services.
Client	<ul style="list-style-type: none">- Génère des requêtes de sélection et de tri complexes sur les données pour effectuer des recherches évoluées sur les fournisseurs de services et filtrer les résultats de ces recherches.

Tableau 8 : vision globale des interactions acteur/base de données

5. Modélisation et conception de la base de données

5.1. Les niveaux de modélisation d'une base de données

La base de données constitue le cœur d'une application mobile. La conception de cette base est la tâche la plus difficile du processus de développement de cette application.

Les méthodes de conception proposent une démarche en étapes et font appel à des modèles pour représenter les objets qui composent le domaine liée à l'application mobile.

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

La modélisation se réalise sur trois niveaux principaux :

- **Niveau conceptuel** : représente le contenu de la base en termes conceptuels, indépendamment de toute considération informatique. Il traduit les concepts du monde réel et les organise selon un schéma appelé schéma conceptuel.
- **Niveau logique** : résulte de la traduction du schéma conceptuel en un schéma propre à un type de bases de données.
- **Niveau physique** : est utilisé pour décrire les méthodes d'organisation et d'accès aux données de la base [14].

La figure 28 illustre les niveaux de modélisation d'une base de données.

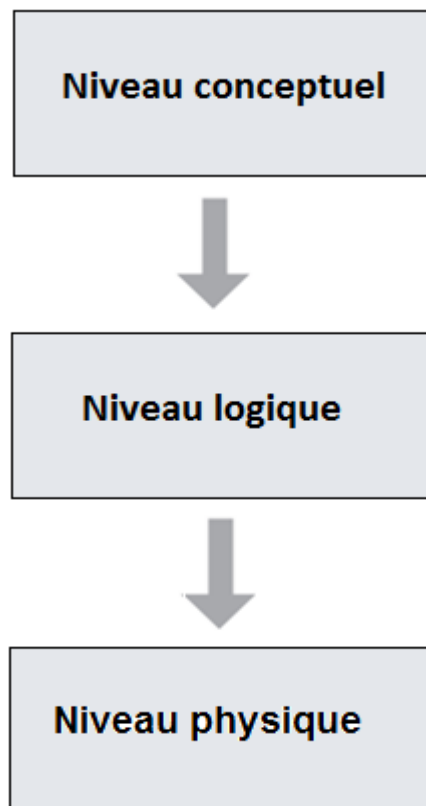


Figure 28 : Niveaux de modélisation d'une base de données

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

Chaque niveau est lié à un modèle de données, qui offre un ensemble de formalismes et de concepts pour représenter les concepts de ce niveau. Le tableau 9 illustre les différents niveaux et modèles associés à chacun.

Niveau	Modèle
Conceptuel	<ul style="list-style-type: none">- Modèle objet- Modèle entité/association (MCD)
Logique	<ul style="list-style-type: none">- Modèle entité/association (MLD)- Modèle relationnel- Modèle objet- Modèle objet relationnel
Physique	<ul style="list-style-type: none">- Modèle entité/association (MPD)- Modèle No SQL

Tableau 9 : Les niveaux de modélisation et les modèles associés

5.2. Méthodologie

Le travail d'analyse réalisé par le premier binôme a permis de modéliser le domaine conceptuel de la base de données en s'appuyant sur le modèle objet. Le résultat de cette analyse est le diagramme de classe d'analyse.

La base de données sous Windev mobile correspond à un Modèle logique de données (MLD) qui peut être généré par L'éditeur d'analyses.

On a commencé notre travail par la réalisation du modèle conceptuel de données (MCD) et du dictionnaire de données à partir du diagramme de classes d'analyse.

La deuxième étape a consisté à générer le modèle logique des données et l'implémenter en utilisant le SGBD relationnel HFSQL client/serveur.

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

La figure 29 illustre d'une manière schématique la méthodologie de modélisation suivie.

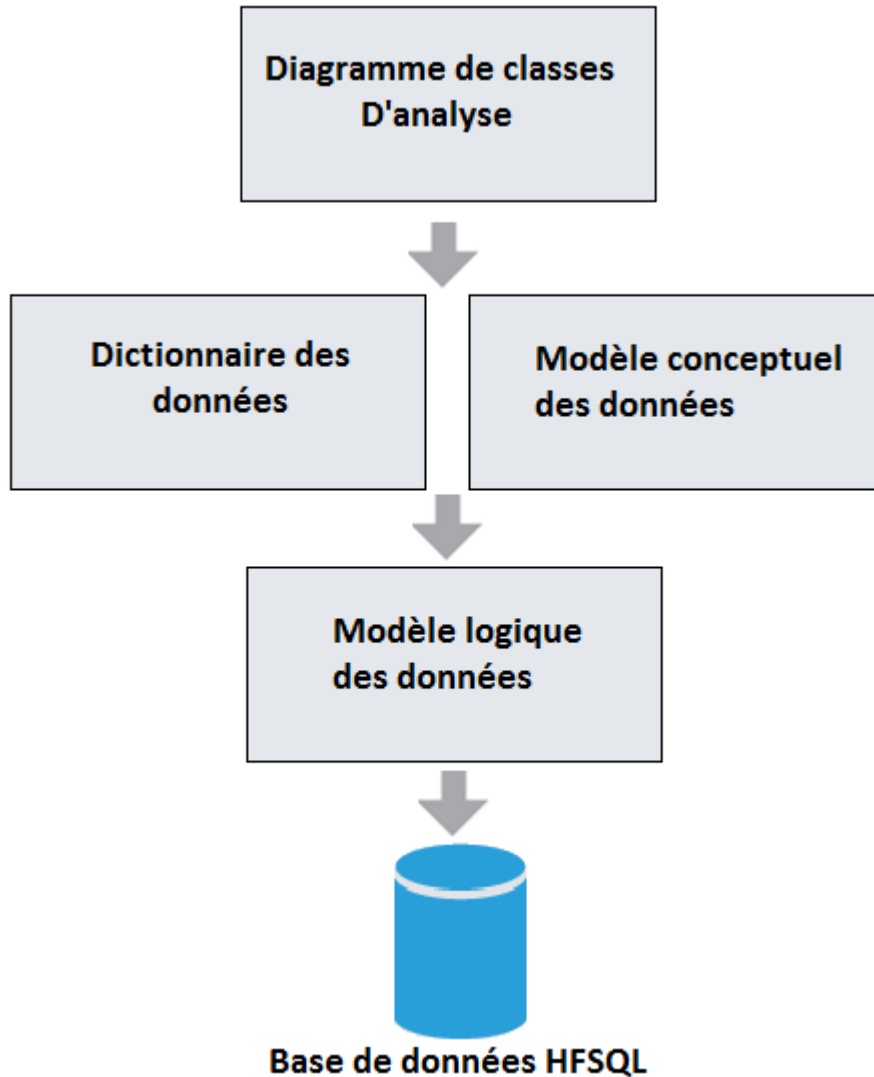


Figure 29 : Méthodologie de modélisation

5.3. Présentation du diagramme de classes d'analyse

La figure 26 représente le diagramme de classes d'analyse issu de l'analyse des besoins réalisée par le premier binôme. Il possède deux classes principales : Utilisateur et publication.

Un utilisateur est identifié de manière unique par un pseudo et il peut poster plusieurs publications mais une publication ne peut être liée qu'à un seul utilisateur.

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

Il y'a deux catégories d'utilisateurs : Client et fournisseur. Un client peut lancer une recherche sur l'ensemble des fournisseurs.

Le fournisseur est soit une personne morale ou soit une personne physique.

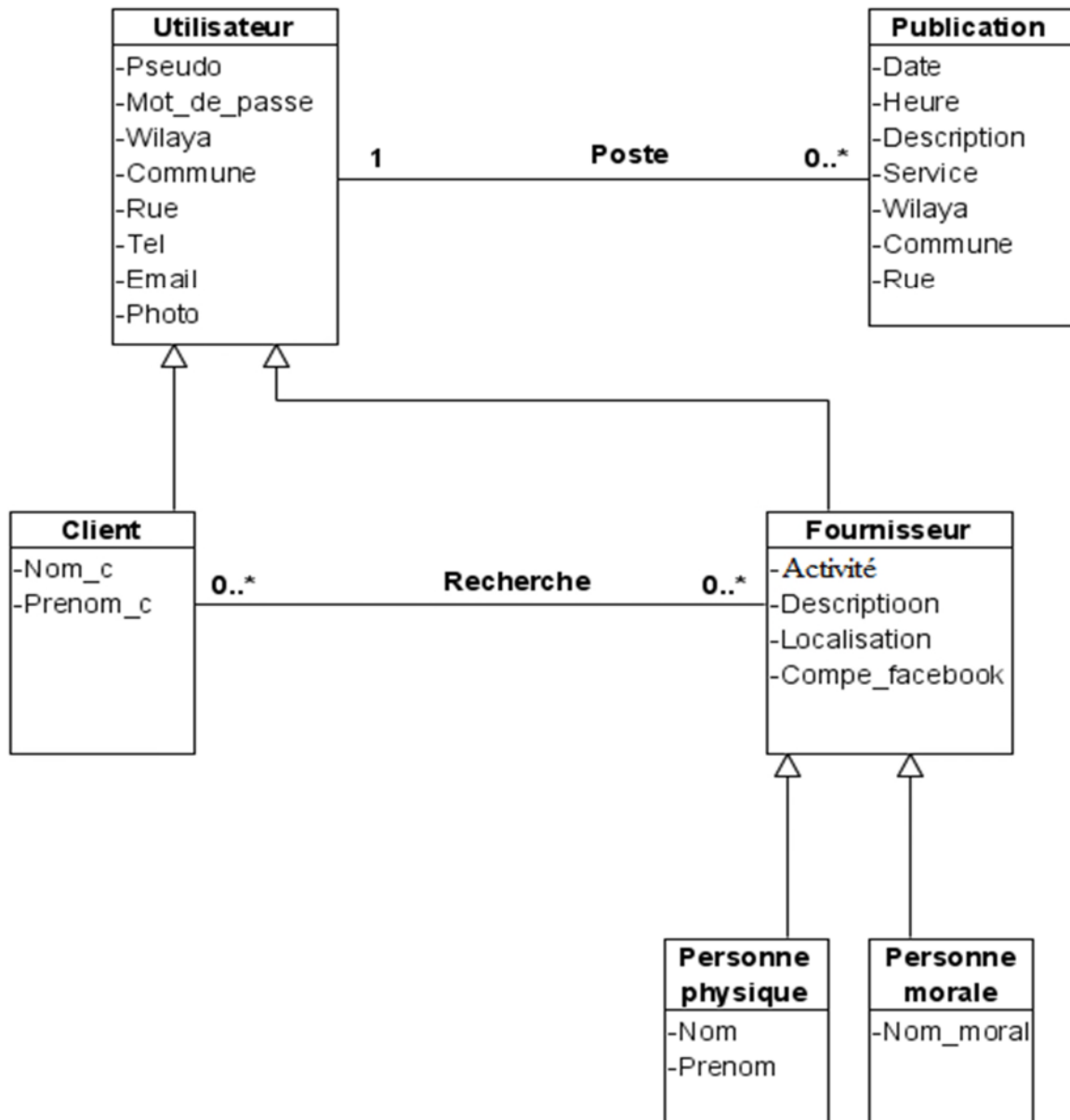


Figure 30: Diagramme de classes d'analyse

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

5.4. Dictionnaire des données

La réalisation du dictionnaire de données est une étape intermédiaire importante dans notre cas, car nous travaillons en équipe sur la même base de données il nous permet de repérer une rubrique facilement dans l'ensemble des données.

Le tableau 10 représente le dictionnaire de données issu du diagramme de classes d'analyse.

Code mnémorique	Désignation	Type	taille
Pseudo_c	Pseudo du client	AN	50
Nom_c	Nom du client	A	50
Prenom_c	Prénom du client	A	50
Mot_de_passe_c	Mot de passe du client	AN	50
Wilaya_c	Wilaya du client	A	50
Commune_c	Commune du client	A	50
Rue_c	Rue du client	AN	80
Telephone_c	Téléphone du client	N	10
Email_c	Email du client	AN	60
Photo_c	Photo du client	Image	
Pseudo_p	Pseudo de la personne physique	AN	50
Nom_p	Nom de la personne physique	A	50
Prenom_p	Prénom de la personne physique	A	50
Mot_de_passe_p	Mot de passe de la personne physique	AN	50
Activité_p	Activité de la personne physique	A	50
Wilaya_p	Wilaya de la personne physique	A	50

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

Commune_p	Commune de la personne physique	A	50
Rue_p	Rue de la personne physique	AN	80
Telephone_p	Téléphone de la personne physique	N	10
Email_p	Email de la personne physique	AN	60
Photo_p	Photo de la personne physique	Image	
Description_p	Description de la personne physique	A	100
Localisation_p	Localisation de la personne physique		
Compte_facebook_p	Compte Facebook de la personne physique	A	50
Pseudo_m	Pseudo de la personne morale	AN	50
Nom_moral	Nom de la personne morale	A	50
Mot_de_passe_m	Mot de passe de la personne morale	AN	50
Activité_m	Activité de la personne morale	A	50
Wilaya_m	Wilaya de la personne morale	A	50
Commune_m	Commune de la personne morale	A	50
Rue_m	Rue de la personne morale	AN	80
Telephone_m	Téléphone de la personne morale	N	10
Email_m	Email de la personne morale	AN	60
Photo_m	Photo de la personne morale	Image	
Description_m	Description de la personne morale	A	100
Localisation_m	Localisation de la personne morale	N	
Compte_facebook_m	Compte Facebook de la personne morale	A	50
IDPublication	Identifiant de la publication	N	

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

Date_pub	Date de la publication	Date	10
Heure_pub	Heure de la publication	Heure	4
Description	Description de la publication	A	100
Activité	Activité liée à la publication	A	50
Wilaya	Wilaya liée à la publication	A	50
Commune	Commune liée à la publication	A	50
Rue	Rue liée à la publication	AN	80

Tableau 10 : Dictionnaire des données

5.5. Le modèle conceptuel des données

Le modèle conceptuel des données est une représentation graphique et structurée des informations mémorisées dans la base de données. Il est basé sur le modèle Entité/association.

Le modèle conceptuel des données est transformé en modèle logique correspondant à la description de la base de données.

Les règles de passage du diagramme de classes au modèle conceptuel des données sont les suivantes :

- **Règle 1:** Chaque classe devient une entité, ses attributs deviennent des propriétés et son attribut clé devient son identifiant. Si une classe ne possède pas d'attribut clé apparent, un identifiant est créé dans l'entité correspondante. Les propriétés créées n'ont pas exactement les mêmes noms que les attributs correspondants, mais ils proviennent du dictionnaire de données.

Exemple : La classe publication devient une entité publication et tous les attributs de la classe deviennent des propriétés. La classe publication n'a pas d'attribut clé alors l'identifiant IDPublication est apparu.

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

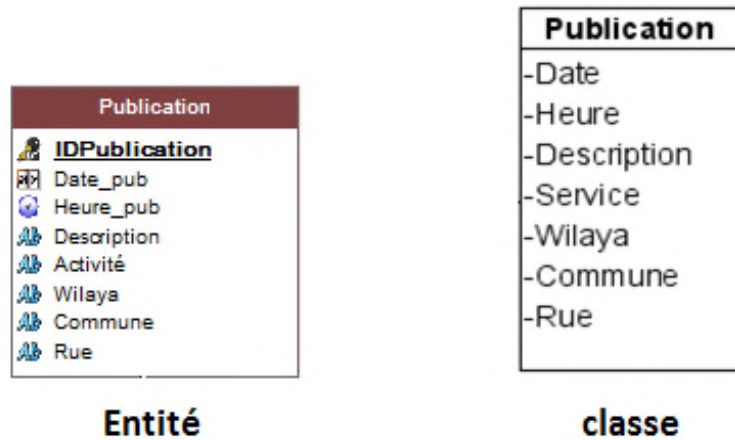


Figure 31 : Exemple de transformation de classe

- **Règle 2 :** Une association est transformée en relation et les cardinalités deviennent des multiplicités des associations mais elles sont notées dans le sens opposé.
- **Règle 3 :** L'héritage n'existe pas dans le MCD, L'entité issue de la superclasse disparaît et toutes les attributs de la superclasse migrent dans la sous classe et deviennent des propriétés de l'entité issue de la sous classe.

Toutes les associations de la superclasse avec d'autres classes deviennent des relations avec l'entité issue de la sous classe.

Exemple :

La classe 'Utilisateur' ne se transforme pas en entité, ses attributs vont migrer vers la classe 'client' qui hérite d'elle. Et l'association 'poste' devient une relation entre 'client' et 'publication'.

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

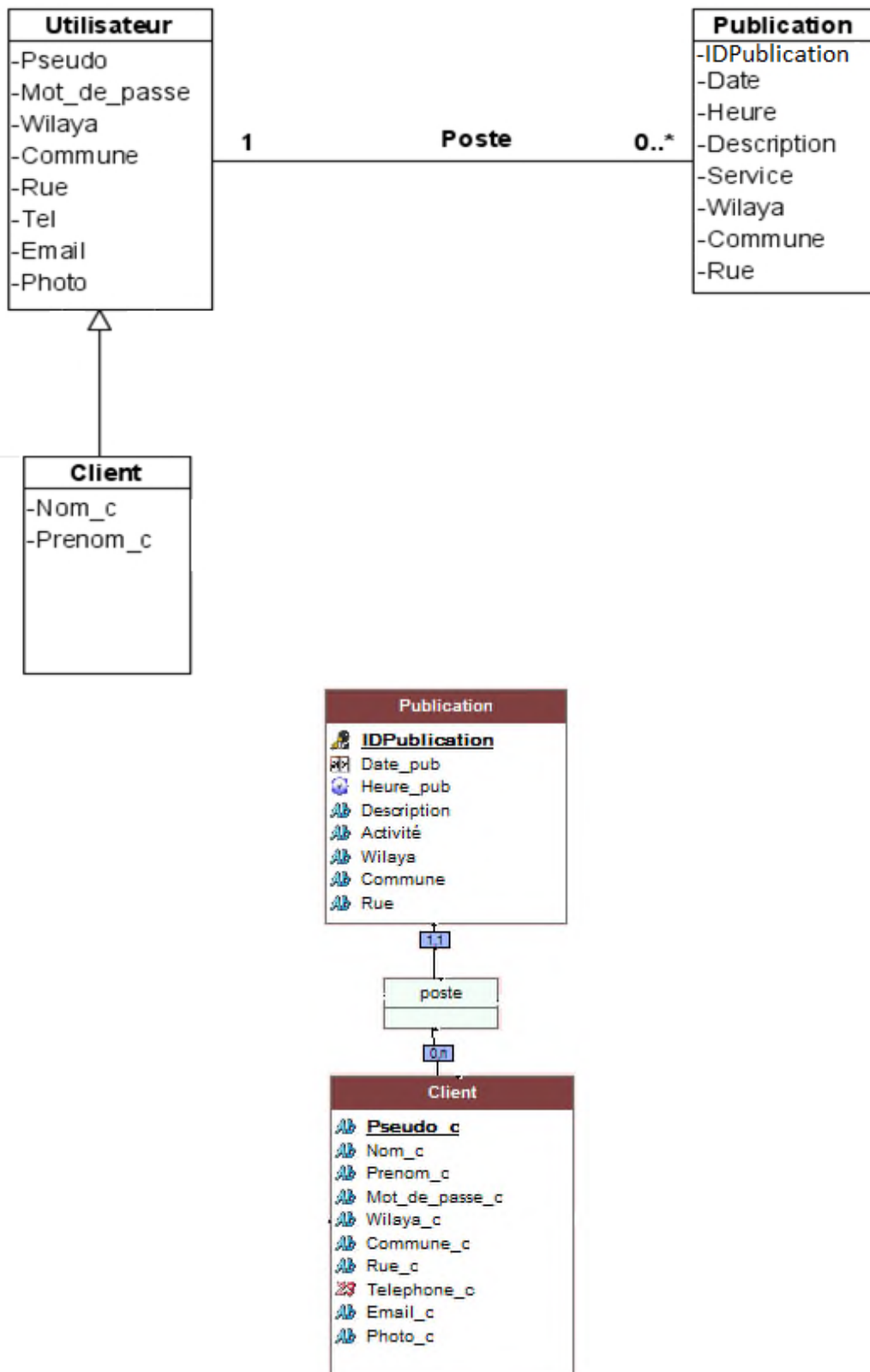


Figure 32 : Exemple de transformation d'héritage

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

La figure 33 représente le modèle conceptuel de données complet créé en utilisant l'éditeur d'analyses de Windev mobile.

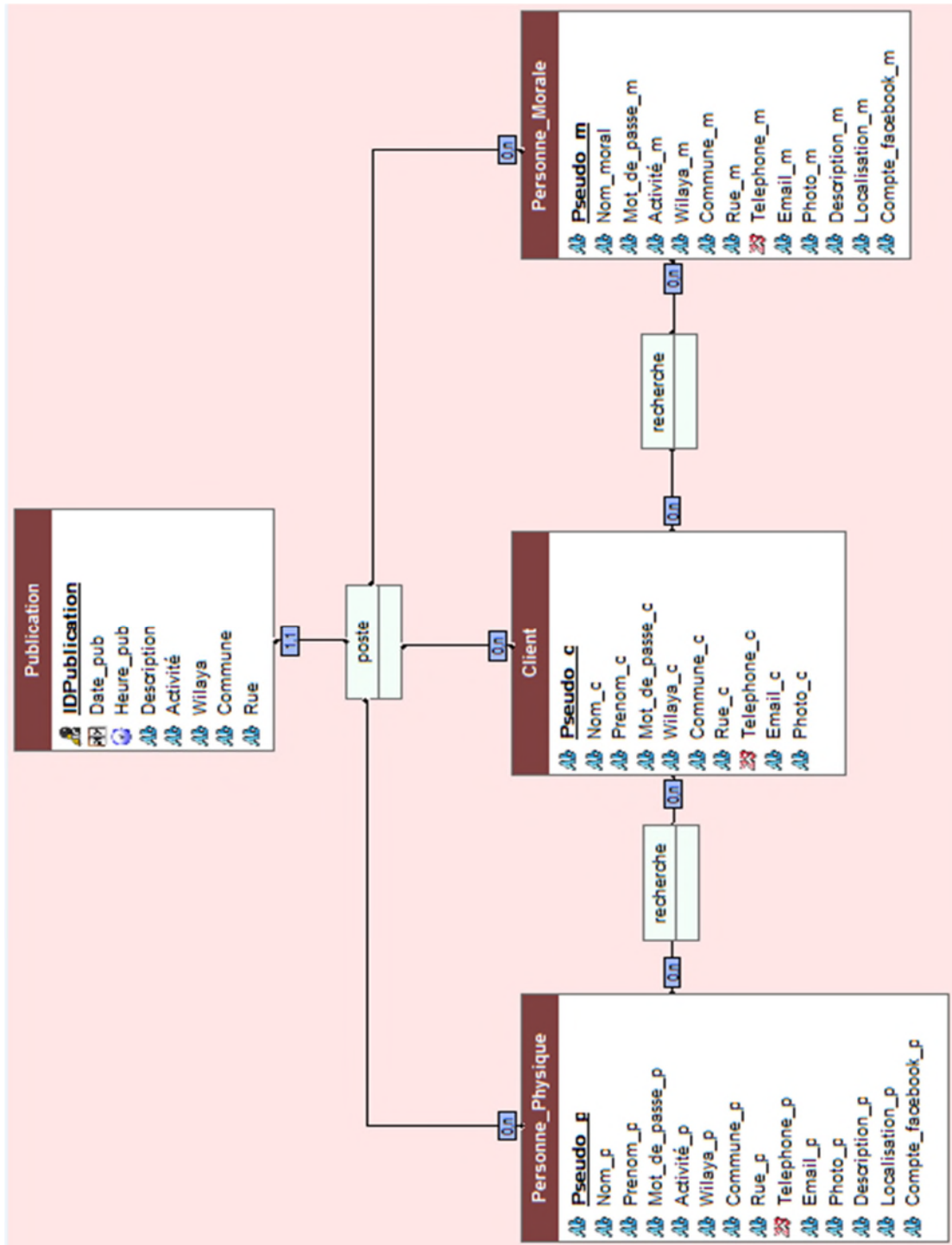


Figure 33 : Modèle conceptuel des données de la base de données

5.6. Le modèle logique des données

Le modèle Logique des données permet de décrire les structures de données, Il correspond au modèle relationnel et pourra être implanté dans une base de données relationnelle.

Le MCD ne peut pas être implanté dans une base de données sans modification et ne peut pas être associé au projet, Il est obligatoire de transformer ce modèle en effectuant un passage au modèle logique de données.

En effet Windev mobile permet de générer le MLD en utilisant l'éditeur d'analyses qui vérifie d'abord la validité du MCD et affiche un rapport d'erreurs si le modèle n'est pas validé.

Avant de présenter le modèle logique généré nous définissons les règles de passage depuis le MCD :

- Règle 1 :

a) Une entité du MCD devient une relation, c'est à dire une table. Chaque ligne correspond aux données d'un objet et chaque colonne correspond à une propriété de cet objet.

b) L'identifiant de l'entité devient la clé primaire de la relation. Les valeurs de la clé primaire sont donc uniques. Les valeurs de la clé primaire sont obligatoirement non nulles.

c) Les autres propriétés deviennent les attributs de la relation.

Dans notre cas les entités "Publication", "Client", "Personne_morale" et "Personne_physique" se transforment aux relations suivantes :

Personne_physique (Pseudo_p, Nom_p, Prenom_p, Mot_de_passe_p, Wilaya_p, Commune_p, Rue_p, Telephone_p, Email_p, Activité_p, Photo_p, Description_p, Compte_facebook_p, Localisation_p).

Personne_morale (Pseudo_m, Nom_moral, Mot_de_passe_m, Wilaya_m, Commune_m, Rue_m, Telephone_m, Email_m, Activité_m, Photo_m, Description_m, Compte_facebook_m, Localisation_m).

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

Client (Pseudo_c, Nom_c, Prenom_c, Mot_de_passe_c, Wilaya_c, Commune_c, Rue_c, Telephone_c, Email_c, Photo_c).

Publication (IDpublication, Date_pub, Heure_pub, Description, Activité, Wilaya, Commune, Rue).

- Règle 2 :

Une association de type 1 : N se traduit par la création d'une clé étrangère dans la relation correspondante à l'entité côté "1".

Dans notre cas l'association "poste" entre l'entité "Publication" et les entités "Client", "Personne_morale" et "Personne_physique" est de type 1 : N. Donc les clés primaires de ces relations migrent dans la relation "Publication" comme clés étrangères.

Publication (IDpublication, Date_pub, Heure_pub, Description, Activité, Wilaya, Commune, Rue, #Pseudo_c, #Pseudo_m, #Pseudo_p).

- Règle 3 :

Une association de type N : N se traduit par la création d'une relation dont la clé primaire est composée des clés étrangères référençant les relations correspondant aux entités liées par l'association. Les éventuelles propriétés de l'association deviennent des attributs de la relation.

Dans notre cas l'association "recherche" entre l'entité "Client" et les entités "Personne_morale" et "Personne_physique" est de type N: N. La relation recherche est donc créée et les clés primaires des trois entités migrent dans la relation comme attributs.

Recherche (Pseudo_c, Pseudo_m)

Recherche1 (Pseudo_c, Pseudo_p)

- Règle 4 :

Une association à trois participants ou plus devient une relation ayant comme clé primaire le composé de toutes les clés primaires des entités participantes à cette relation.

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

Dans notre cas l'association "poste" entre l'entité "Publication" et les entités "Client", "Personne_morale" et "Personne_physique" devient une relation ayant comme clé primaire l'ensemble des clés primaires "Pseudo_p", "Pseudo_m" et "Pseudo_c".

poste (IDPublication, Pseudo_m'', Pseudo_p'', Pseudo_c'')

Le modèle logique des données complet de la base de données est le suivant :

Personne_physique (Pseudo_p, Nom_p, Prenom_p, Mot_de_passe_p, Wilaya_p, Commune_p, Rue_p, Telephone_p, Email_p, Activité_p, Photo_p, Description_p, Compte_facebook_p, Localisation_p).

Personne_morale (Pseudo_m, Nom_moral, Mot_de_passe_m, Wilaya_m, Commune_m, Rue_m, Telephone_m, Email_m, Activité_m, Photo_m, Description_m, Compte_facebook_m, Localisation_m).

Client (Pseudo_c, Nom_c, Prenom_c, Mot_de_passe_c, Wilaya_c, Commune_c, Rue_c, Telephone_c, Email_c, Photo_c).

Publication (IDpublication, Date_pub, Heure_pub, Description, Activité, Wilaya, Commune, Rue, #Pseudo_c, #Pseudo_m, #Pseudo_p).

Recherche (Pseudo_c, Pseudo_m)

Recherche1 (Pseudo_c, Pseudo_p)

poste (IDPublication, Pseudo_m'', Pseudo_p'', Pseudo_c'')

Avant la génération du MLD l'éditeur d'analyse permet de l'implanter automatiquement en utilisant n'importe quelle SGBD et en utilisant n'importe quel type d'accès : natif, ODBC ou OLE DB.

Nous avons généré le MLD lié à notre projet en utilisant HFSQL client/serveur

La figure 34 représente le modèle logique visuel généré par L'éditeur d'analyses de Windev mobile.

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

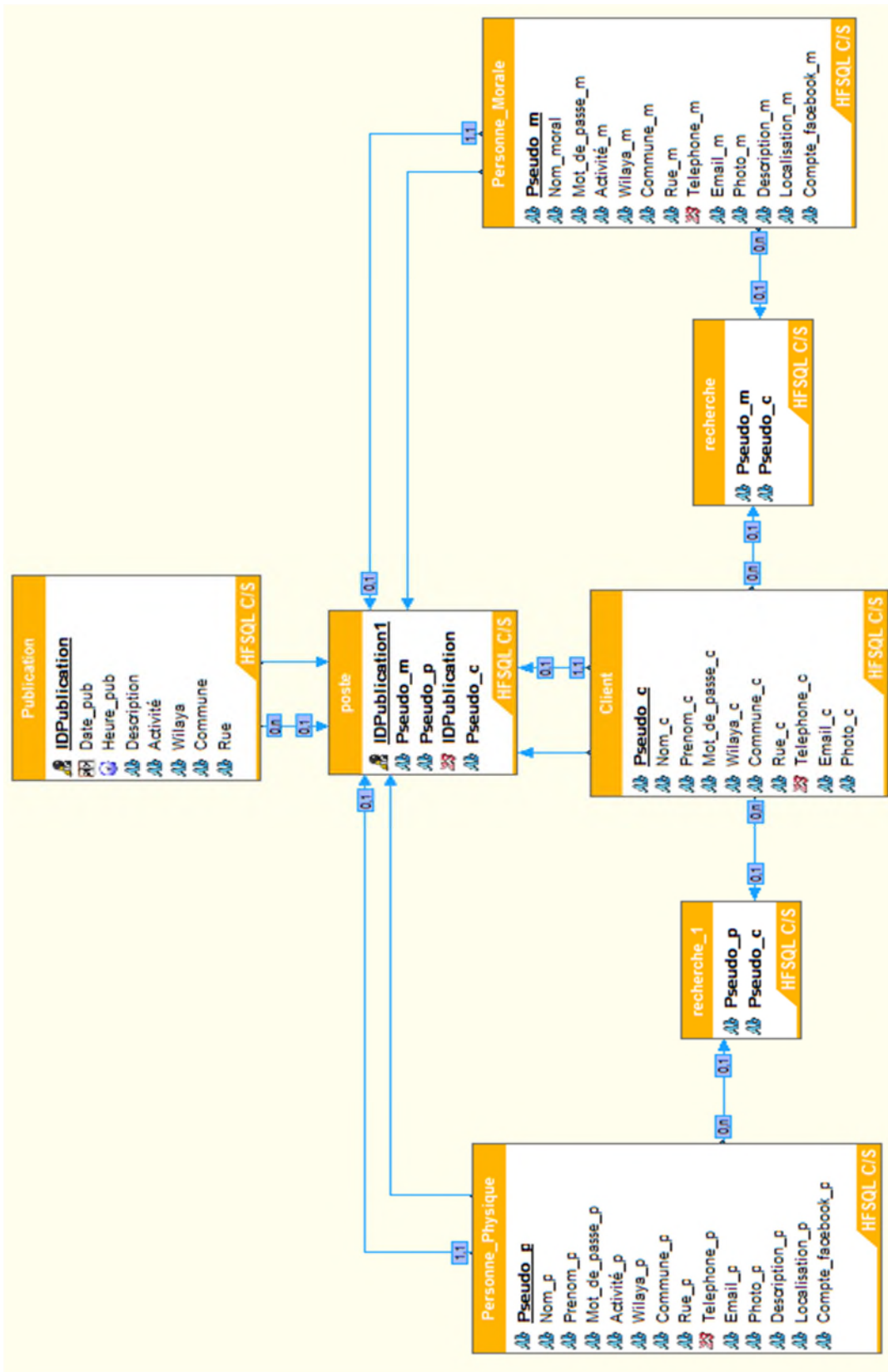


Figure 34 : Modèle logique des données généré par Windev mobile

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

5.7. Code SQL correspondant à la création de la base de données

```
-- Création de la table Client
CREATE TABLE `Client` (
  `Pseudo_c` VARCHAR(50) NOT NULL UNIQUE ,
  `Nom_c` VARCHAR(50) NOT NULL ,
  `Prenom_c` VARCHAR(50) NOT NULL ,
  `Mot_de_passe_c` VARCHAR(50) NOT NULL ,
  `Wilaya_c` VARCHAR(50) NOT NULL ,
  `Commune_c` VARCHAR(50) NOT NULL ,
  `Rue_c` VARCHAR(80) NOT NULL ,
  `Telephone_c` INTEGER NOT NULL ,
  `Email_c` VARCHAR(60) NOT NULL ,
  `Photo_c` LONGBLOB NOT NULL );
```

Figure 35 : code SQL pour la création du fichier de données client

```
-- Création de la table Personne_Morale
CREATE TABLE `Personne_Morale` (
  `Pseudo_m` VARCHAR(50) NOT NULL UNIQUE ,
  `Nom_moral` VARCHAR(50) NOT NULL ,
  `Mot_de_passe_m` VARCHAR(50) NOT NULL ,
  `Activité_m` VARCHAR(50) NOT NULL ,
  `Wilaya_m` VARCHAR(50) NOT NULL ,
  `Commune_m` VARCHAR(50) NOT NULL ,
  `Rue_m` VARCHAR(80) NOT NULL ,
  `Telephone_m` INTEGER NOT NULL ,
  `Email_m` VARCHAR(60) NOT NULL ,
  `Photo_m` LONGBLOB NOT NULL ,
  `Description_m` VARCHAR(100) NOT NULL ,
  `Localisation_m` LONGBLOB NOT NULL ,
  `Compte_facebook_m` VARCHAR(50) NOT NULL );
```

Figure 36 : code SQL pour la création du fichier de données Personne_morale

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

```
-- Création de la table Personne_Physique
CREATE TABLE `Personne_Physique` (
  `Pseudo_p` VARCHAR(50) NOT NULL UNIQUE ,
  `Nom_p` VARCHAR(50) NOT NULL ,
  `Prenom_p` VARCHAR(50) NOT NULL ,
  `Mot_de_passe_p` VARCHAR(50) NOT NULL ,
  `Activité_p` VARCHAR(50) NOT NULL ,
  `Wilaya_p` VARCHAR(50) NOT NULL ,
  `Commune_p` VARCHAR(50) NOT NULL ,
  `Rue_p` VARCHAR(80) NOT NULL ,
  `Telephone_p` INTEGER NOT NULL ,
  `Email_p` VARCHAR(60) NOT NULL ,
  `Photo_p` LONGBLOB NOT NULL ,
  `Description_p` VARCHAR(100) NOT NULL ,
  `Localisation_p` LONGBLOB NOT NULL ,
  `Compte_facebook_p` VARCHAR(50) NOT NULL );
```

Figure 37 : code SQL pour la création du fichier de données Personne_physique

```
-- Création de la table poste
CREATE TABLE `poste` (
  `Pseudo_c` VARCHAR(50) NOT NULL ,
  `IDPublication` INTEGER NOT NULL ,
  `Pseudo_p` VARCHAR(50) NOT NULL ,
  `Pseudo_m` VARCHAR(50) NOT NULL ,
  `IDPublication1` INTEGER PRIMARY KEY );
```

Figure 38 : code SQL pour la création du fichier de données poste

```
-- Création de la table Publication
CREATE TABLE `Publication` (
  `IDPublication` INTEGER PRIMARY KEY ,
  `Date_pub` DATE NOT NULL ,
  `Heure_pub` TIME NOT NULL ,
  `Description` VARCHAR(100) NOT NULL ,
  `Activité` VARCHAR(50) NOT NULL ,
  `Wilaya` VARCHAR(50) NOT NULL ,
  `Commune` VARCHAR(50) NOT NULL ,
  `Rue` VARCHAR(80) NOT NULL );
```

Figure 39 : code SQL pour la création du fichier de données Publication

CHAPITRE 3

Etude de cas- Base de données d'une application mobile pour la mise en relation de services-

```
-- Création de la table recherche
CREATE TABLE `recherche` (
  `Pseudo_c` VARCHAR(50) NOT NULL ,
  `Pseudo_m` VARCHAR(50) NOT NULL );
```

Figure 40 : code SQL pour la création du fichier de données recherche

```
-- Création de la table recherche_1
CREATE TABLE `recherche_1` (
  `Pseudo_c` VARCHAR(50) NOT NULL ,
  `Pseudo_p` VARCHAR(50) NOT NULL );
```

Figure 41 : code SQL pour la création du fichier de données recherche1

Conclusion

La première partie de ce chapitre a mis l'accent sur la démarche adoptée pour la réalisation du projet et la deuxième a été consacrée à la modélisation et la conception de la base de données de l'application mobile "Ypiresia".

Dans le chapitre suivant nous allons aborder les interactions de l'application avec la base de données et les tests réalisés, Ainsi que l'environnement matériel et les technologies utilisés.

CHAPITRE IV :
Réalisation et Evaluation

Introduction

La partie précédente a mis l'accent sur la modélisation de la base de données. Cette dernière partie sera consacrée à l'implémentation des différentes fonctionnalités attendues et la mise en interaction avec la partie applicative de l'application.

1. Matériel utilisé

Pour pouvoir relier notre base de données à un serveur, Nous avons installé le serveur HFSQL client/serveur sur deux machines :

- Un PC portable DELL
 - Processeur Intel Core I7 2,2 GHz.
 - 8 Go de mémoire vive.
 - Disque dur de capacité 1 To.
 - Système d'exploitation Microsoft Windows 7 64 bits.

- Un PC portable HP
 - Processeur Intel Core I3 2,40 GHz.
 - 4 Go de mémoire vive.
 - Disque dur de capacité 500 Go.
 - Système d'exploitation Microsoft Windows 7 64 bits.

Pour tester l'interaction de notre base de données avec un terminal mobile, Nous avons utilisé ces deux appareils :

- Une tablette Samsung Galaxy Tab 2
 - Processeur 2CPU 1,60 GHz.
 - 512 Mo de mémoire vive.
 - Capacité de stockage 5 Go.
 - Système d'exploitation Android JELLY BEAN 4.1.1.

- Un smartphone Lenovo A319
 - Processeur 2CPU 1,30 GHz.
 - 512 Mo de mémoire vive.
 - Capacité de stockage 4 Go.
 - Système d'exploitation Android JELLY BEAN 4.4.2.

2. Fonctionnement de la base de données

Notre base de données est de type HFSQL client/server, une fois déployé sur un serveur HFSQL, elle utilise le protocole RPC pour communiquer avec les terminaux mobiles. Voici les différentes étapes pour établir une communication :

- 1- Le client lance une requête en utilisant la fonction HExécuteRequête.
- 2- Le client exécute automatiquement la fonction HConnecteAccèsDistant qui permet d'ouvrir l'analyse.
- 3- La librairie WDXXXCOM.DLL envoie une requête RPC au serveur RPC.
- 4- La librairie WDXXXCOM.DLL présente sur le serveur HFSQL reçoit la requête.
- 5- La librairie WDXXXCOM. transmet les ordres SQL à la librairie HFSQL (WDXXXHF.DLL).
- 6- La librairie WDXXXHF. Ouvre l'analyse présente sur le poste serveur.
- 7- La librairie WDXXXHF Renvoie la réponse à la librairie WDXXXCOM.DLL du serveur RPC.
- 8- La librairie WDXXXCOM.DLL envoie la réponse RPC au client.

Si on prend comme exemple notre base de données appelée ypiresia, elle correspond à au fichier ''ypiresia.WDD''.

Si un client cherche un fournisseur appelé ''Tarik'', l'application lance une requête de sélection dans la table Personne_morale en utilisant la fonction HExécuteRequête comme suit :

```
HExécuteRequête (SELECT * FROM Personne_morale WHERE  
Nom_moral='Tarik')
```

La librairie WDXXXCOM.DLL envoie une requête de type RPC vers le serveur en utilisant la fonction HConnecteAccèsDistant.

CHAPITRE 4

Réalisation et évaluation

Le code SQL est comme suit :

```
HConnecteAccèsDistant ("192.168.1.1", "Admin", "root", "C:\ypiresia.WDD",  
"Mot de passe")
```

- 192.168.1.1 est l'adresse du serveur
- Admin est le nom d'utilisateur
- Root est le mot de passe
- Ypiresia.WDD est le fichier physique de la base de données ypiresia

Le serveur envoie la réponse au client.

La figure suivante représente le processus de communication entre une base de données et le client mobile.

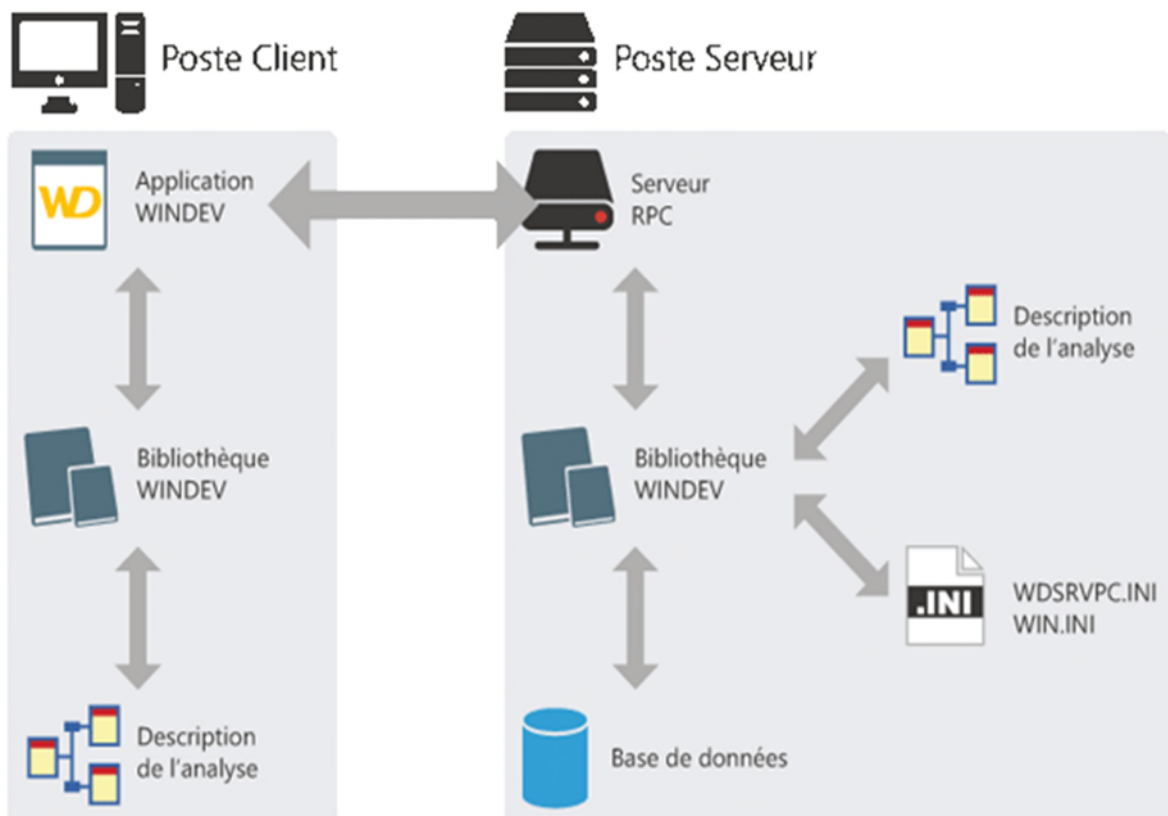


Figure 42 : Fonctionnement de la base de données de l'application Ypiresia

3. Captures d'écran relatives aux différentes opérations sur la base de données

Voici quelques captures d'écrans pour les fenêtres principales qui effectuent des opérations sur la base de données



Figure 43 : fenêtre correspondante à la vérification d'un enregistrement

L'utilisateur saisie le pseudo et le mot de passe, dès qu'il clique sur le bouton se connecter, le code WLangage associé au bouton va lancer une requête de vérification dans les tables 'Client', 'Personne_physique' et 'Personne_morale'. Si les champs mot de passe et pseudo saisis correspondent aux champs Mot_de_passe et Pseudo_c ou Pseudo_m et Pseudo_p il y'a authentification.

CHAPITRE 4

Réalisation et évaluation

Ce code WLangage implémente la vérification de l'appartenance d'un enregistrement à la base de données.

```
HLitRecherchePremier(client,pseudo_c,SAI_pseudoo)
HLitRecherchePremier(personne_morale,pseudo_m,SAI_pseudoo)
HLitRecherchePremier(personne_phisique,pseudo_p,SAI_pseudoo)
SI (HTrouve(client) ET (client.password=SAI_passee)) ALORS
  Utilise(FEN_profil_client2)
  SINON
SI (HTrouve(personne_morale) ET (personne_morale.password=SAI_passee)) ALORS
  Utilise(FEN_profil_fournisseur_société2)
  SINON
SI (HTrouve(personne_phisique) ET (personne_phisique.password=SAI_passee)) ALORS
  Utilise(FEN_profil_fournisseur)
  SINON
SI SAI_pseudoo<>"" OU SAI_passee<>"" ALORS
  Erreur("pseudo ou mot de passe sont incorrecte ")
  SINON
SI SAI_pseudoo="" OU SAI_passee="" ALORS
  Erreur("un champ non saisie ")
  FIN
  FIN
  FIN
  FIN
  FIN
```

Fenêtre 44 : code WLangage qui implémente la vérification d'un enregistrement

L'utilisateur saisie le pseudo et le mot de passe. Une fois le bouton se connecter est cliqué, la fonction HLitRecherchePremier lance une recherche dans les tables 'Client', 'Personne_phisique' et ' Personne_morale ' et se positionne sur le premier résultat trouvé. Par exemple HLitRecherchePremier (Client, pseudo_c, SAI_pseudoo) permet de chercher dans la table client les enregistrements qui ont pseudo_c identique au pseudo saisi par l'utilisateur.

La fonction HTrouve vérifie si l'enregistrement correspond à la recherche en cours.

La fonction Utilise permet d'ouvrir la fenêtre fournie en paramètre et fermer la fenêtre courante.

SAI_passee correspond au champ de saisi du mot de passe et SAI_pseudoo correspond au champ de saisie du pseudo.

CHAPITRE 4

Réalisation et évaluation

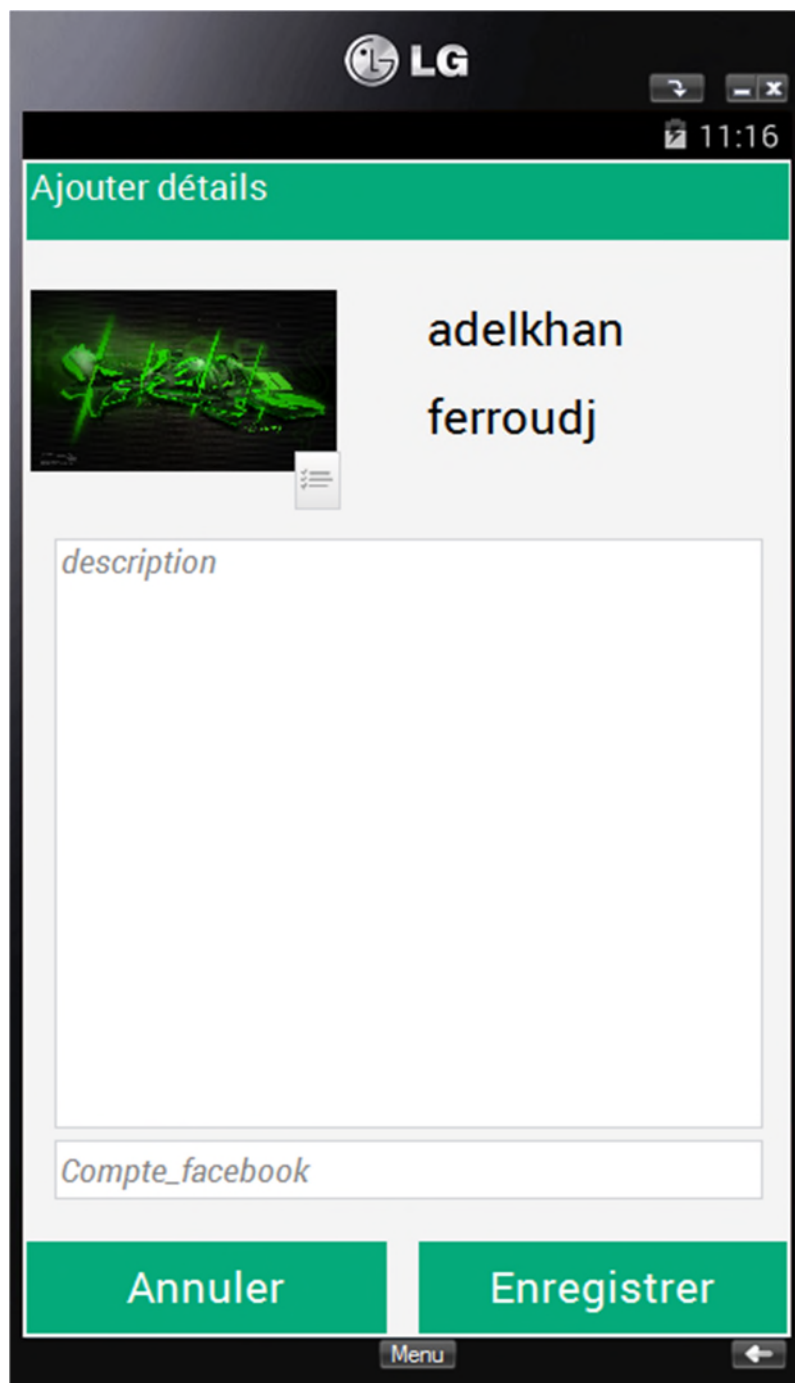


Figure 45 : Fenêtre correspondante à la mise a jour des informations du client

Le fournisseur de service saisi des nouvelles informations sur son profil, une fois le bouton Enregistrer est cliqué une requête de mise à jour de l'enregistrement correspondant est envoyé à la base de données.

CHAPITRE 4

Réalisation et évaluation

Le code suivant implémente la mise à jour des informations du client.

```
EcranVersFichier()  
HModifie(client, hNumEnrEnCours)
```

Figure 46 : Code WLangage de la mise à jour des informations du client

La fonction EcranVersFichier permet de transférer les informations saisies dans la fenêtre dans la base de données et La fonction HModifie de les modifier.

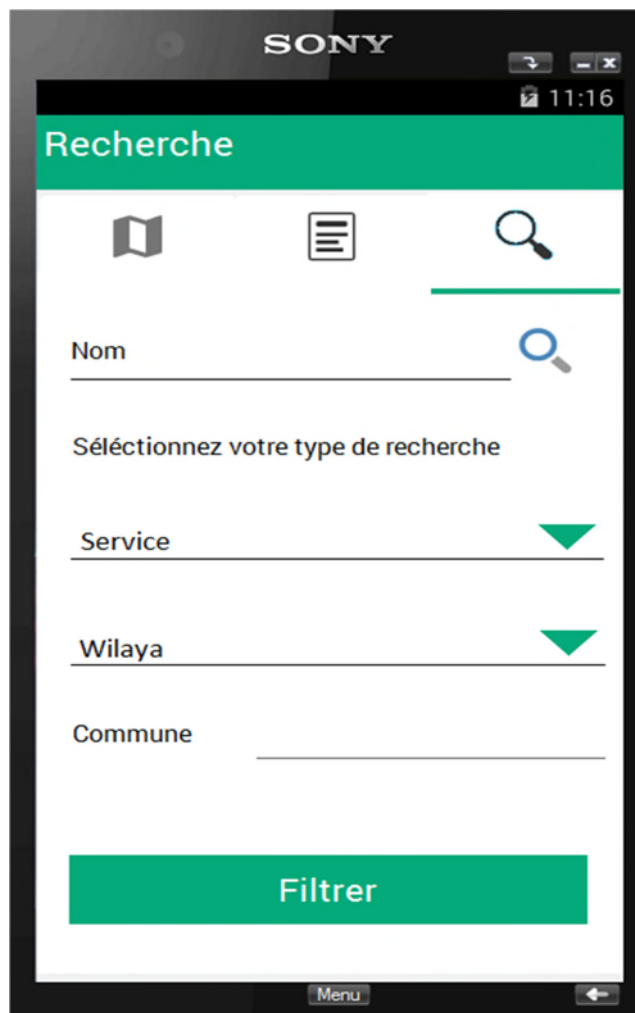


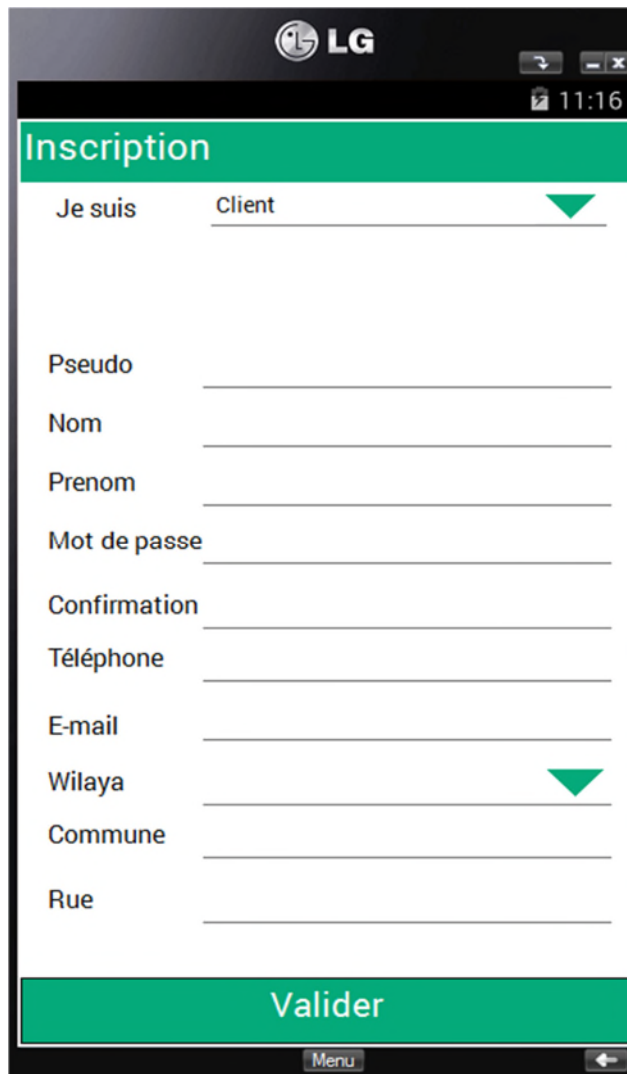
Figure 47 : Fenêtre correspondante à la recherche d'un fournisseur

CHAPITRE 4

Réalisation et évaluation

Voici un code WLangage qui nous a permis d'implémenter la recherche

```
HActiveFiltre (Personne_morale)
HLitRecherchePremier (Personne_morale, service, ONG_profil_client.COMBO_service)
HFilterIdentique (personne_morale, service, COMBO_service)
Si HTrouve(Personne_morale)=Vrai alors
OuvreFenêtreMobile(FEN_Resulta_de_recherche_société)
HLitSuivant (personne_morale, service)
SINON
OuvreFenêtreMobile (fen_Pas_de_résultats)
HRAZ(Personne_morale)
HDésactiveFiltre(Personne_morale)
Fin
```



The image shows a mobile application interface for registration. At the top, there is a header bar with the LG logo and the time 11:16. Below the header, the title 'Inscription' is displayed in a green bar. The form consists of several fields: 'Je suis' with a dropdown menu set to 'Client', 'Pseudo', 'Nom', 'Prenom', 'Mot de passe', 'Confirmation', 'Téléphone', 'E-mail', 'Wilaya' with a dropdown menu, 'Commune', and 'Rue'. A green button labeled 'Valider' is at the bottom. A 'Menu' button is visible at the very bottom of the screen.

Figure 48 : Fenêtre correspondante à l'insertion d'un enregistrement dans la table *Personne_physique*

CHAPITRE 4

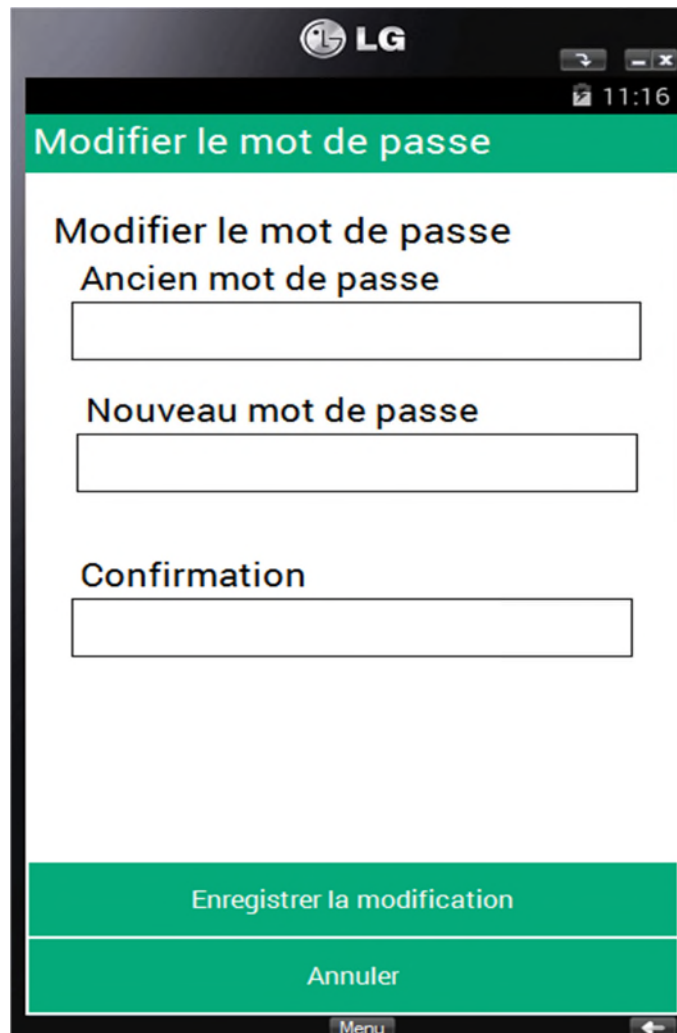
Réalisation et évaluation

Le client remplit tous les champs et clique sur le bouton valider, ce qui va générer une requête d'insertion d'un nouvel enregistrement dans la table 'Client'

Le code WLangage suivant permet d'envoyer la requête.

```
SI SAIS_password=SAIS_con_pass ALORS  
  EcranVersFichier()  
  HAjoute(client)  
  Utilise(FEN_profil_client2)  
FIN
```

Figure 49 : Code WLangage pour l'insertion d'un enregistrement dans la table client



The screenshot shows a mobile application interface for password modification. The title bar at the top is green and contains the text 'Modifier le mot de passe'. Below the title bar, the main content area is white and contains the following elements:

- A title 'Modifier le mot de passe' in bold black text.
- A label 'Ancien mot de passe' above a white rectangular input field.
- A label 'Nouveau mot de passe' above a white rectangular input field.
- A label 'Confirmation' above a white rectangular input field.
- At the bottom, there are two green buttons with white text: 'Enregistrer la modification' and 'Annuler'.

The application is running on an LG device, as indicated by the logo in the top left corner. The time 11:16 is displayed in the top right corner. A 'Menu' button is visible at the bottom center of the screen.

Figure 50 : Fenêtre correspondante à la mise à jour de la valeur du champ mot de passe

CHAPITRE 4

Réalisation et évaluation

L'utilisateur saisie l'ancien et le nouveau mot de passe et il clique sur le bouton Enregistrer la modification. Une requête de mise à jour du champ Mot_de_passe est générée.

Le code WLangage suivant permet d'envoyer une requête de type modification du champ Mot_de_passe_m vers la table Personne_morale.

```

[ SI SAI_anc_passe="" OU SAI_nouv_pass="" ALORS
  Info("un champ non saisie ")
SINON
  HlitRecherchePremier(personne_morale,password,SAI_anc_passe)
[ SI (HTrouve(personne_morale)=Vrai) ET (personne_morale.password=SAI_anc_passe) ALORS
  EcranVersFichier()

  [ SI HModifie(personne_morale)=Vrai ALORS
    ToastAffiche("Enregistrement effectué", toastCourt, cvMilieu, chCentre)

  ]
]
FIN
SINON
  Erreur("verifiez votre ancien mot de passe")
FIN
]
FIN
```

Figure 51 : code WLangage pour la modification du champ mot de passe

4. Test de la base de données

Le test de la base de données est la dernière étape de son cycle de vie, son objectif est assurer son bon fonctionnement.

Pour tester la base de données il faut :

- Tester la connexion au serveur
- Tester les requêtes vers la base.

1- Test de la connexion au serveur :

Avant de tester la connexion il faut la créer et la paramétrer, le but principal de Ce test est d'optimiser les performances d'accès aux fichiers physiques et le fonctionnement en cas de pannes ou d'arrêts imprévisibles.

CHAPITRE 4

Réalisation et évaluation

La création de la connexion se fait par assistant du centre de contrôle HFSQL ou en utilisant un code WLangage.

Voici le code qu'on a utilisé pour créer notre base de données ypiresia

```
MaConnexion2 est une Connexion

// Paramètres de la connexion
MaConnexion2..Provider = hAccèsHFClientServeur
MaConnexion2..Utilisateur = "admin"
MaConnexion2..MotDePasse = ""
MaConnexion2..Serveur = "127.0.0.1"
MaConnexion2..BaseDeDonnées = "Ypiresia_BDD"

// Ouverture de la connexion
HOuvreConnexion(MaConnexion2)
HChangeConnexion("", MaConnexion2)
```

Figure 52 : code WLangage pour connecter la base de données ypiresia au serveur HFSQL

La connexion est une variable de type Connexion.

Provider correspond au mode d'accès, Admin est le nom d'utilisateur par défaut, Le mot de passe est vide, Le serveur HFSQL est installé sur la machine locale, son adresse est 127.0.0.1 et le nom de la base de données est Ypiresia_BDD.

Les fonctions HOuvreConnexion et HChangeConnexion sont utilisé pour ouvrir et changer dynamiquement de connexion en cas d'arrêts non prévus.

Il est également facile de créer une connexion en utilisant le centre de contrôle HFSQL e n précisant les mêmes paramètres.

Nous avons testé la connexion au serveur HFSQL en utilisant le déboguer présent sur l'environnement Windev Mobile.

2- Test des requêtes :

Après le test de la connexion les tests des requêtes est indispensable pour optimiser les résultats des requêtes et les temps d'exécution.

CHAPITRE 4

Réalisation et évaluation

Nous avons testé l'ensemble des requêtes créées en utilisant l'outil de test proposé par l'éditeur de requêtes.

Voici un exemple de requête qu'on a testé

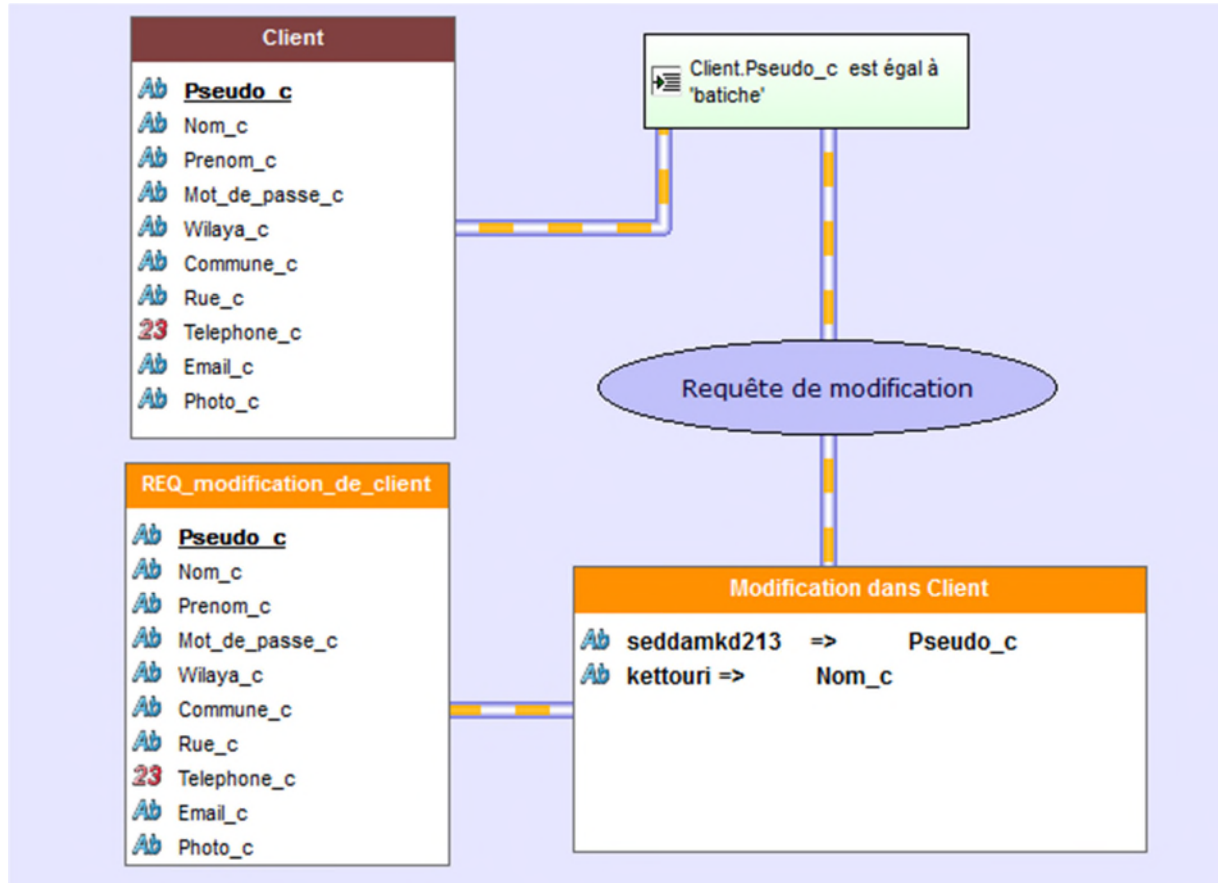


Figure 53 : test d'une requête de modification

La requête créée est une modification dans la table `Client` de l'enregistrement qui a pour clé primaire `Pseudo_c = 'batiche'`, `Pseudo_c` remplacé par `seddamkd213` et `Nom_c` par `kettouri`.

Conclusion

Cette dernière partie nous a permis de situer et évaluer notre travail, nous avons exposés le matériel utilisé, le fonctionnement de notre base de données et son interaction avec l'IHM de l'application réalisée.

Enfin nous avons montré quelques tests réalisés sur notre base de données.

Conclusion générale

Notre projet d'étude qui a porté sur la conception et réalisation d'une base de données pour une application mobile pour la mise en relation du consommateur et du fournisseur de service sur l'AGL Windev mobile.

Avec comme objectif principal de définir la meilleure solution afin d'implanter une application mobile capable d'aider les consommateurs et les fournisseurs de services à communiquer conformément à l'expression de leurs besoins. Cela étant il s'avère indispensable de donner une conclusion par rapport au projet réalisé. Notre travail s'est focalisé sur la spécification des besoins, la modélisation d'une base de données et la transition de la phase d'analyse vers la phase de conception qui a permis la mise en place d'une solution conceptuelle robuste, évolutive, maintenable et facile à réaliser sur l'environnement Windev Mobile.

Ce projet nous à permis d'avoir une approche complète du développement d'un logiciel et d'appliquer nos acquis en génie logiciel.

Perspectives

Dans l'ensemble, nous pouvons signaler, avec sincérité que nous n'avons pas atteint la perfection, mais ça n'empêche pas que notre projet reste prometteur et ambitieux.

Nous avons pour ambition d'enrichir l'application "ypiresia", l'améliorer soit sur l'aspect fonctionnel soit sur l'aspect ergonomique et enfin la publier sur le Market Android.

Nous recommandons aussi aux autres étudiants d'apporter leurs changements pour que ce travail soit plus attractif.

Glossaire

IEEE: Institute of Electrical and Electronics Engineers

Association professionnelle, l'IEEE est constituée d'ingénieurs électriciens, d'informaticiens, de professionnels du domaine de télécommunications, etc. L'organisation a pour but de promouvoir la connaissance dans le domaine de l'ingénierie électrique (électricité et électronique).

ANSI : American National Standards Institute

Organisation privé à but non lucratif qui supervise le développement de normes pour les produits, les services, les procédés, les systèmes et les employés des États-Unis.

Test unitaire :

En programmation informatique, le test unitaire est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »).

Dictionnaire de données :

Collection de données nécessaire à la conception d'une base de données relationnelle. Cours sur <http://www.hotosting.com/cresite/>. C'est un référentiel de vocabulaire commun de l'organisation. Il constitue la base sur laquelle s'appuient les décisions et stratégies d'une entreprise.

SGBD : Système de Gestion de Bases de Données

Logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Debugger :

Programme d'aide à l'élimination des erreurs (bogues) dans un logiciel.

Unix :

Système d'exploitation multitâches et multiutilisateur créé en 1969 par Kenneth Thompson, conceptuellement ouvert et fondé sur une approche par laquelle il offre de nombreux petits outils chacun dotés d'une mission spécifique.

Make :

Logiciel qui construit automatiquement des fichiers, souvent exécutables, ou des bibliothèques à partir d'éléments de base tels que du code source. Il utilise des fichiers appelés makefile qui spécifient comment construire les fichiers cibles.

Smalltalk :

Est un langage de programmation orienté objet, réflexif et dynamiquement typé. Il fut l'un des premiers langages de programmation à disposer d'un environnement de développement intégré.

Titanium Appcelerator :

Kit de construction de logiciels (Framework open-source) destinés aux téléphones mobiles utilisant Android ou IOS.

Adobe Flash Builder :

Contrôle ActiveX, un plugin ou un lecteur multimédia autonome utilisant la technique flash.

Smartphone: terme utilisé pour désigner les terminaux téléphoniques possédants des fonctions proches d'un ordinateur personnel grâce à un système d'exploitation évolué. Les fonctions les plus courantes concernant l'accès internet, le courrier électronique, la gestion d'emploi du temps et du carnet d'adresses. Exemple iPhone d'Apple.

App Store :

Plateforme de téléchargement d'applications similaire au Google Play, distribué par Apple sur les appareils mobiles fonctionnant sous IOS (iPod Touch, iPhone et iPad)

Google Play :

Boutique en ligne créé par Google en 2012, par fusion des services Android Market, Google Movies, Google ebookstore et Google Music.

Windows Store :

Boutique en ligne créé par Microsoft, Microsoft fusionne le Windows Store et le Windows Phone afin d'unifier Smartphones, tablettes et PC.

iTunes :

Logiciel de lecture et de gestion de bibliothèque multimédia numérique distribué gratuitement par Apple.

Mac OS X :

Ligne de systèmes d'exploitation propriétaire développés et commercialisés par Apple.

Phonegap :

Framework open source pour créer des applis mobiles multiplateforme avec les technologies traditionnelles du web : HTML, CSS, JavaScript.

Objective-c :

Langage de programmation orienté objet réflexif. C'est une extension du C ANSI, comme le C++, mais qui se distingue de ce dernier par sa distribution dynamique des messages, son typage faible ou fort, son typage dynamique et son chargement dynamique. Contrairement au C++, il ne permet pas l'héritage multiple mais il existe toutefois des moyens de combiner les avantages de C++ et d'Objective-c.

XCode :

Environnement de développement pour Mac OS X, ainsi que pour IOS, watchOS et tv OS

Framework :

En programmation informatique, un Framework est un kit de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).

Cocoa Touch :

Est une API native d'Apple pour le développement orienté objet sur son système d'exploitation Mac OS X.

WinJS :

Bibliothèque JavaScript, c'est-à-dire un ensemble de fichiers qui contiennent des fonctions, des classes et des objets divers qui vont aider les développeurs dans le code.

Pattern :

Arrangement caractéristique de modules, reconnue comme bonne pratique en réponse à un problème de conception d'un logiciel. Il décrit une solution standard, utilisable dans la conception de différents logiciels.

API : Application Programming Interface

Interface pour langages de programmation, matérialisées par des primitives, permettant à une application d'accéder à des programmes système pour, par exemple, communiquer ou extraire des données. Ainsi un jeu développé avec l'API de Direct 3D fonctionnera avec les cartes 3D supportant cette interface, sans adaptation.

SOAP : Simple Object Access Protocol

Protocole standard de communication. SOAP est un protocole décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes. Il circule sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance.

JQuery :

Bibliothèque de JavaScript libre et multiplateforme créée pour faciliter l'écriture des scripts côté client dans le code HTML des pages web.

JSON : Java Script Object Notation

Un format de données textuelles, générique, dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple. Créé par Douglas Crock Ford entre 2002 et 2005, il est décrit par la RFC4627 de l'IETF.

Cloud :

Cloud computing ou l'informatique en nuage ou encore l'infonuagique, est l'exploitation de la puissance de calcul ou de stockage de serveurs informatiques distants par l'intermédiaire d'un réseau, généralement internet. Ses serveurs sont loués à la demande, le plus souvent par tranche d'utilisation selon des critères techniques (puissance, bande passante, etc.) mais également au forfait.

OLE DB (parfois orthographié OLEDB ou OLE-DB) :

OLE DB est une API développée par Microsoft permettant l'accès aux données.

OLE DB se sert d'interfaces COM (Component Object Model). Il a été conçu dans le but de remplacer ODBC, de ce fait il permet l'accès à des bases de données exotiques ou des sources de données qui n'utilisent pas un processeur de requêtes SQL.

WDMMap :

WDMMap est un utilitaire permettant de visualiser et de manipuler un fichier de données lors du développement d'une application.⁷

WDHFDiff :

WDHFDiff est un utilitaire permettant de comparer des fichiers de données HFSQL.

DLL : Dynamic Link Library

DLL un ensemble de fonctions qui sont chargées en mémoire par un programme, au besoin, lors de son exécution.

PDA :

Un PDA (Personal Digital Assistant, littéralement assistant numérique personnel, aussi appelé organisateur) est un ordinateur de poche composé d'un processeur, de mémoire vive, d'un écran tactile et de fonctionnalités réseau dans un boîtier compact d'extrêmement petite taille.

Liste des abréviations

- RIM**: Research In Motion.
- ANSI**: American National Standards Institute.
- HTML**: **H**yper Text Markup Langage.
- CSS**: **C**ascading Style Sheets.
- API**: **A**pplication Programming Interface.
- UML**: **U**nified Modeling Langage.
- IHM**: **I**nterface Homme Machine.
- ODBC**: **O**pen database Connectivity.
- SQL**: **S**tructured Query Langage.
- HFSQL**: Hyper File Structured Query Langage.
- MCD** : Modèle Conceptuel de Données.
- MLD** : Modèle Logique de Données.
- MPD** : Modèle Physique de Données.
- OLEDB**: Object Linking and Embedding, Database.
- MDAC**: Windows Data Access Components.
- **RPC**: Remote procedure call.
- DLL** : Dynamic Link Library.

Références bibliographiques

- [1] : Lydie du Bousquet, Processus de développement, cycles de vies.
- [2] : Cours GL Master2, université Abderrahmane Mira, Bejaia.
- [3] : Fabrice Huet, Outils de génie logiciel.
- [4] : Majdi blaghbi, Génie logiciel, Institut Supérieur des Etudes Technologiques de Djerba.
- [5] : [http// :www.developpez.com](http://www.developpez.com).
- [6] : [http// :www.blog.erlem.fr](http://www.blog.erlem.fr).
- [7] : Synertic, conception et développement d'applications pour smartphones et tablettes.
- [8] : PC SOFT, Windev Mobile 21, concepts, 01-10-2015.
- [9] : Livre des fonctionnalités WinDev Version WD170 – 0512.
- [10] : [http// :www.pcsoft.fr/Documentation](http://www.pcsoft.fr/Documentation).
- [11]: PC SOFT, Cours Windev, id5187.
- [12]: [http://:www.pcsoft.fr/Documentation](http://www.pcsoft.fr/Documentation).
- [13]: PC SOFT, Cours Windev, id5195.

Références bibliographiques

[1]:Lydie du Bousquet. Processus de développement, cycles de vie.2007.Repéré à: http://membreslig.imag.fr/dubousquet/docs/2.2_CyclesDeVie.pdf. Consulté le 14/05/2016.

[2]:Cours Génie logiciel, Chapitre 1.2016.Université Abderrahmane Mira Bejaïa.

[3]:Fabrice Huet. Outils du génie logiciel.2003. Repéré à:https://www.google.fr/?gws_rd=ssl#q=Fabrice+huet+outils+de+g%C3%A9nie+logiciel.Consulté le 15/05/2016.

[4]:Majdi belaghi. Cours universitaire. Génie logiciel.2006. Repéré à:<http://www.isetjb.rnu.tn/docs/divers/ouvrages-iset-djerba-2015%20.pdf>. Consulté le 15/05/2016.

[5]:Les environnements de développement [Article].Repéré à: <http://general.developpez.com/edi.html> .Consulté le 18/05/2016.

[6]:Le développement logiciel.[Article]. Repéré à: <http://www.blog.erlem.fr/programmation/edi.html>. Consulté le 19/05/2016.

[7]:Synertic. Conception et développement d'applications pour Smartphones et tablettes [Article]. Repéré à:http://www.synertic.fr/sites/default/files/pdf/synertic%20Mobile_v2.pdf. Consulté le 21/05/2016.

[8]:PC SOFT. E-book Windev mobile 21, concepts. Repéré à: <http://www.pcsoft.fr/windevmobile/concepts.html>. Consulté le 28/05/2016.

[9]:PC SOFT. Livre des fonctionnalités de Windev. Repéré à: http://www.pcsoft-windev-webdev.com/Fonctionnalites_WinDev.pdf.

Consulté le 05/06/2016.

[10]:PC SOFT. Documentation officielle de Windev mobile. Repéré à:
<http://doc.pcssoft.fr>. Consulté le 18/06/2016.

[11]:E-book meilleur cours de Windev. Repéré à:
<http://wind.developpez.com/tutoriels>. Consulté le 23/06/2016.

[12]: PC SOFT. Documentation officielle de Windev mobile. Repéré à:
<http://doc.pcssoft.fr>. Consulté le 23/06/2016.

[13]:Tutoriel WinDev mobile. Repéré à: <http://www.cours-gratuit.com/cours-winddev/cours-winddev-complet-en-pdf>. Consulté le 24/06/2016.

ANNEXE

A.1 Connexion à une base de données HFSQL

Pour que l'application puisse manipuler les fichiers de données présents sur le serveur HFSQL, il est nécessaire de définir dans le projet la connexion qui sera utilisée par l'application. Cette connexion peut être définie :

- soit depuis l'éditeur d'analyses.
- soit par programmation.

Il est également possible de permettre à une application d'utiliser au choix une base de données HFSQL Client/serveur ou une base de données HFSQL Classic.

Les fichiers de données accédés par un serveur HFSQL ne sont accessibles que via ce serveur. Ils ne peuvent pas être accédés directement par une autre application notamment à l'aide du moteur HFSQL Classic. Il est nécessaire de se connecter au serveur pour accéder à ces fichiers de données.

Définition des connexions depuis l'éditeur d'analyses

Pour définir une connexion depuis l'éditeur d'analyses :

1. Sous le volet "Analyse", dans le groupe "Connexion", cliquer sur "Nouvelle connexion". L'assistant de définition d'une nouvelle connexion se lance.
2. Conserver l'option "HFSQL Client/serveur" et passez au plan suivant de l'assistant.
3. Dans les différents écrans de l'assistant, saisir les informations demandées :
 - nom du serveur,
 - numéro du port réseau,
 - nom et mot de passe de l'utilisateur,
 - base de données,
 - nom et libellé de la connexion.
4. Valider l'assistant.

Une boîte de dialogue s'affiche et propose d'associer des fichiers de données à cette nouvelle connexion. Accepter.
5. Sélectionner les fichiers de données à associer et valider. Il est possible de copier directement les fichiers de données existants sur le serveur HFSQL. Cette opération peut être réalisée si nécessaire ultérieurement.
6. Afficher la liste des connexions : sous le volet "Analyse", dans le groupe "Connexion", cliquer sur "Connexions".
 - Tester si nécessaire cette nouvelle connexion : cliquer sur le bouton "Tester".
 - Les différents onglets permettent de visualiser et de modifier les caractéristiques des connexions :
 - Onglet "Propriétés" : caractéristiques de la connexion définies dans l'assistant (nom, libellé, type, ...)

- Onglet "Avancé" : Numéro de port de la connexion, options de compression et de cryptage des données de la connexion.
Si les fichiers de données utilisant la connexion sont cryptés, il est conseillé de crypter également la connexion.

- Onglet "Utilisée par" : Liste des fichiers de données utilisant la connexion.

7. Fermer la fenêtre des connexions définies.

En exécution, la connexion associée au fichier de données manipulé sera automatiquement ouverte lors du premier accès à ce fichier de données.

Les fichiers de données associés à une connexion de type "HFSQL Client/Serveur" se transforment automatiquement en fichiers de données de type "HFSQL Client/Serveur".

Pour les reconnaître, une indication "HFSQL/CS" apparaît au bas de la description graphique de ces fichiers. De plus, la couleur associée à cette zone devient orangée :



Crypter une connexion

Pour crypter une connexion :

1. Afficher la description de la connexion : sous le volet "Analyse", dans le groupe "Connexion", cliquer sur "Connexions".
2. Sélectionner la connexion à crypter.
3. Afficher l'onglet "Avancé".
4. Sélectionner le mode de cryptage : Rapide ou RC5 (16 boucles).
5. Valider.

Définition des connexions par programmation

Pour définir une connexion par programmation, plusieurs méthodes peuvent être utilisées :

- Utilisation des fonctions [HDécritConnexion](#) ou [HOuvreConnexion](#).
- Utilisation du type de données [Connexion](#) et de ses propriétés.

Définir une connexion à l'aide des fonctions HDécritConnexion ou HOuvreConnexion

Pour définir une connexion à l'aide des fonctions HDécritConnexion ou HOuvreConnexion :

1. Définir la connexion grâce à la fonction [HDécritConnexion](#) ou [HOuvreConnexion](#).

Le paramètre <Source des données> doit correspondre au nom ou à l'adresse IP du serveur à utiliser.

Si le port réseau à utiliser est différent du port par défaut (port 4900), le paramètre <Source des données> doit être de la forme :

"<NomOuAdresseIPServeur>:<NuméroPort>".

Le paramètre <Provider OLEDB ou Accès Natif> doit correspondre à la constante *hAccèsHFClientServeur*.

Par exemple :

```
HDécritConnexion("MaConnexion", "Julie", "MotDePasse", "MonServeurHF",  
, ...
```

```
    "MaBaseDeDonnées", hAccèsHFClientServeur, hOLecture)
```

ou

```
HouvreConnexion("ConnexionServeur", "Stephane", "", "ServeurDonnées:540  
0", ...
```

```
    "MesDonnées", hAccèsHFClientServeur)
```

2. Ouvrir la connexion grâce à la fonction HOuvreConnexion.

Cette étape n'est pas nécessaire si la connexion a été définie et ouverte à l'étape précédente grâce à la fonction **HOuvreConnexion**.

3. Associer la connexion aux différents fichiers de données grâce à la fonction HChangeConnexion.

Par exemple :

```
HChangeConnexion(Client, "MaConnexion")
```

ou, pour associer tous les fichiers de l'analyse à la connexion :

```
HChangeConnexion("", "ConnexionServeur")
```

Remarque : Pour crypter la connexion, il suffit de spécifier des informations étendues spécifiques ("CRYPTAGE = RAPIDE" ou "CRYPTAGE = RC5_16". Pour plus de détails, consulter la documentation des fonctions **HDécritConnexion** ou **HOuvreConnexion**.

Définir une connexion à l'aide du type Connexion

Pour définir une connexion à l'aide du type Connexion :

1. Définir la connexion grâce au type connexion et à ses propriétés.

Par exemple :

```
MaConnexion est une Connexion  
// Description de la connexion  
MaConnexion..Utilisateur = "USER"  
MaConnexion..MotDePasse = "PASSWORD"  
MaConnexion..Serveur = "MONSERVEUR"  
MaConnexion..BaseDeDonnées = "Base de données"  
MaConnexion..Provider = hAccèsHFClientServeur  
MaConnexion..Accès = hOLectureEcriture  
MaConnexion..InfosEtendues = "Infos étendues"  
MaConnexion..OptionsCurseur = hCurseurClient
```


2. Ouvrir la connexion grâce à la fonction **HOuvreConnexion**.

Par exemple :

```

MaConnexion est une Connexion
// Description de la connexion
MaConnexion..Utilisateur = "USER"
MaConnexion..MotDePasse = "PASSWORD"
MaConnexion..Serveur = "MONSERVEUR"
MaConnexion..BaseDeDonnées = "Base de données"
MaConnexion..Provider = hAccèsHFClientServeur
MaConnexion..Accès = hOLectureEcriture
MaConnexion..InfosEtendues = "Infos étendues"
MaConnexion..OptionsCurseur = hCurseurClient

```

A.2 Extensions des fichiers manipulés par Windev

Extension	Description	Thème	Ne doit pas être supprimé
.AWL	Fichier généré pour chaque page d'un site dynamique. Correspond à la bibliothèque de la page. Utilisé lors d'un déploiement page à page .	Exécution	X
.AWP	Page WEBDEV AWP	Exécution	X
.AWWS	Bibliothèque contenant toutes les procédures d'un Webservice. Ces procédures seront utilisables par n'importe quelle application utilisant votre Webservice. Pour plus de détails, consultez Générer un Webservice .	Exécution	
.BKC	Fichier archive d'une classe (créé à la première sauvegarde de la classe)	Archive	
.BKE	Fichier archive d'un état (créé à la première sauvegarde de l'état)	Archive	
.BKG	Fichier archive d'une collection de procédures (créé à la première sauvegarde de la collection de procédures)	Archive	
.BKH	Fichier archive d'un système d'aide (créé à la première sauvegarde du système d'aide)	Archive	
.BKP	Fichier archive du projet (créé à la première sauvegarde du projet)	Archive	
.BKR	Fichier archive d'une requête (créé à la	Archive	

	première sauvegarde de la requête)		
.BKT	Fichier archive d'un modèle créé à la première sauvegarde (modèle de fenêtres , modèle de champs ou modèle de pages).	Archive	
.BKW	Fichier archive d'une fenêtre WINDEV (créé à la première sauvegarde de la fenêtre)	Archive	
.CPD	Charte de programmation	Éditeurs	X
.DEP	Liste des dépendances d'un projet	Outils WINDEV	
.DIC	Description du dictionnaire	Éditeurs	X
.DPL	Duplicata d'impression	Exécution	X
.DRT	Table des droits du groupware développeur .	Éditeurs	X
.EMF	Image générée pour les aperçus sous les éditeurs.	Éditeurs	X
.ENV	Paramètres de l' environnement de l'éditeur et des projets utilisés	Éditeurs	
.ERR	Messages d'erreur de chaque composante	Éditeurs	X
.EXE	Fichier exécutable	Exécution	
.FIC	Fichier de données HFSQL	HFSQL	X
.FTX	Fichier correspondant à un index full-text .	HFSQL	X
.GAB	Gabarit de WINDEV/WEBDEV	Éditeurs	X
.HSP	Procédure stockée HFSQL	HFSQL	X
.HTM	Page HTML statique	Exécution	X
.LST	Fichier contenant le nom des différents fichiers HFSQL à sauvegarder (créé par WDJournal)	Outils	
.MCD	Description du Modèle Conceptuel des Données(MCD)	Analyse	X
.MDE	Procédures des modèles du RAD	RAD	X
.MDF	Fichier interne de l'analyse	Analyse	X
.MDL	Fichier de description des modèles du RAD(pattern)	RAD	X
.MGR	Partie graphique de l'analyse	Analyse	X

.MLD	Description du Modèle Logique des Données(MLD)	Analyse	X
.MMO	Fichier de mémo HFSQL	HFSQL	X
.NDX	Fichier d'index HFSQL	HFSQL	X
.NET	WUpdate.NET (poste des utilisateurs finaux) Fichier contenant les références des différentes applications à contrôler lors d'une installation réseau et, pour chacune d'entre elles, les caractéristiques du contrôle à effectuer. Ce fichier est créé par WDADMINEXE	Outils	X
.PATTERN	Fichier correspondant au pattern RAD .	Éditeurs	X
.RAD	Informations RAD de l'analyse	Analyse	X
.REP	Correspondance entre les fichiers logiques et les fichiers physiques manipulés par une application. Pour plus de détails, consultez la Gestion du fichier ".REP"	HFSQL	
.REW	URL Rewriting	Exécution	X
.RPA	Fichier journal contenant les informations de réplication . Ce fichier est échangé entre la base de données maître et la base de données abonnée.	Outils	X
.RPL	Description des réplicas abonnés	Outils	X
.RPM	Description des réplicas maître	Outils	X
.RPP	Paramètres de la réplication	Outils	X
.SEM	Fichier interne de l'analyse.	Analyse	X
.SQL	Requête enregistrée avec WDSQL	Outils	X
.SRP	Scénario créé et utilisé par WDRepl pour effectuer une réplication	Outils	X
.STY	Description des styles des champs	Éditeurs	X
.SVG	Format d'image pouvant être généré.	Exécution	
.SYN	Informations de synchronisation de la réplication universelle	Outils	X
.TKC	Fichier archive d'une classe (créé à chaque	Archive	

	sauvegarde de la classe)		
.TKE	Fichier archive d'un état (créé à chaque sauvegarde de l'état)	Archive	
.TKG	Fichier archive d'une collection de procédures (créé à chaque sauvegarde de la collection de procédures)	Archive	
.TKH	Fichier archive d'un système d'aide (créé à chaque sauvegarde du système d'aide)	Archive	
.TKP	Fichier archive du projet (créé à chaque sauvegarde du projet)	Archive	
.TKR	Fichier archive d'une requête (créé à chaque sauvegarde de la requête)	Archive	
.TKW	Fichier archive d'une fenêtre (créé à chaque sauvegarde de la fenêtre)	Archive	
.TRS	Journal des opérations en transaction	HFSQL	X
.TRX	Journal des valeurs avant la transaction	HFSQL	X
.UML	Modèle UML	Éditeurs	X
.VER	Fichier interne de l'analyse.	Analyse	X
.WAMB	Ambiance WEBDEV	Éditeurs	X
.WBC	Liste des identificateurs des classes Attention : En WINDEV 5.5 , cette extension correspondait aux informations de compilation d'une classe.	Compilation	
.WBE	Liste des identificateurs des états Attention : En WINDEV 5.5 , cette extension correspondait aux informations de compilation d'un état.	Compilation	
.WBG	Liste des identificateurs des collections de procédures	Compilation	
.WBL	Compte-rendu de batches multi-projets	Éditeurs	
.WBP	Liste des identificateurs d'un projet Attention : En WINDEV 5.5 , cette extension correspondait aux informations de compilation d'un projet.	Compilation	

.WBR	Liste des identificateurs d'une requête .	Compilation	
.WBW	Liste des identificateurs des fenêtres Attention : En WINDEV 5.5 , cette extension correspondait aux informations de compilation d'une fenêtre.	Compilation	
.WCC	Informations de compilation d'une classe (recréé à chaque compilation)	Compilation	
.WCE	Informations de compilation d'un état (recréé à chaque compilation)	Compilation	
.WCG	Informations de compilation d'une collection de procédures (recréé à chaque compilation)	Compilation	
.WCI	Liste des éléments du composant interne .	Éditeurs	
.WCP	Informations de compilation d'un projet (recréé à chaque compilation)	Compilation	
.WCW	Informations de compilation d'une fenêtre (recréé à chaque compilation)	Compilation	
.WDA	Description de l' analyse et des fichiers HFSQL (en édition) pour l'éditeur d'analyses	Éditeurs	X
.WDAP	Sauvegarde du fichier WDA de la dernière analyse générée	Éditeurs	
.WDAUDIT	Fichier correspondant à un audit dynamique	Éditeurs	
.WDBATCH	Fichier Batch	Éditeurs	X
.WDC	Classe	Éditeurs	X
.WDD	Description de l' analyse et des fichiers HFSQL (en exécution)	Éditeurs	X
.WDE	Etat	Éditeurs	X
.WDF	Fichier de paramètres pour l' installation personnalisée d'un projet	Éditeurs	X
.WDFAB	Bibliothèque de plans d'action	Éditeurs	X
.WDG	Collection de procédures Attention : En WINDEV 5.5 , cette extension correspondait au fichier contenant la liste des fichiers utilisés par WDInst .	Éditeurs	X
.WDH	Description du système d'aide	Éditeurs	X

.WDI	Description d'un composant externe (uniquement pour le développement d'applications)	Composant externe	X
.WDIT	Fichier de traduction d'un composant externe	Composant externe	X
.WDK	Fichier contenant tous les éléments d'un composant externe	Composant externe	X
.WDL	Bibliothèque	Exécution	X
.WDM	Messages d'erreur des différentes DLL du framework. Ce fichier est généré par WDInt (outil de traduction des messages d'erreur)	Éditeurs / Outils	X
.WDO	Liste des différents éléments livrés avec un composant externe (fichiers de données, fichiers .INI, ...).	Composant externe	X
.WDP	Projet WINDEV	Éditeurs	X
.WDR	Requête Attention : En WINDEV 5.5 , cette extension correspondait au module "include" des fichiers de données	Éditeurs	X
.WDT	Modèle de fenêtres, Modèle de champs	Éditeurs	X
.WDU	Nom et chemin complet des éléments intégrés à une bibliothèque	Éditeurs	
.WDUMP	Dump de débogage	Exécution	
.WDV	Description des conversions de fichier (généré par WDConver ou par l'éditeur d'analyses)	Outils	X
.WDW	Fenêtre WINDEV	Éditeurs	X
.WDY	Fichier Gabarit en exécution. Ce fichier est nécessaire pour appliquer un gabarit dynamiquement (fonction ChangeGabarit). Si le projet est associé à un gabarit, le fichier WDY correspondant au gabarit est automatiquement intégré à la WDL (bibliothèque).	Éditeurs	X
.WDZ	Fichier archive (créé par exemple avec WDZip)	Éditeurs	X
.WEM	Modèle souple	Éditeurs	X

.WEP	Erreurs de compilation du projet	Compilation	
.WER	Description de réplication	Outils	X
.WGP	Fichier interne de compilation	Compilation	
.WIG	Fichier interne de compilation	Compilation	
.WL	Code WLangage pouvant être intégré dans une application ou un site (par le mot-clé EXTERNE) Quatre fichiers sont fournis en standard avec WINDEV (répertoire \personnel\Externe) : "WinConst.wl" : Constantes standard de Windows. "Limites.wl" : Constantes correspondant aux limites des types de données WINDEV . "ListeDéfinitionHF.wl" : Constantes HFSQL utilisées pour la journalisation . "EXCEPT.wl" : Codes d'exception les plus courants (à récupérer avec la fonction ExceptionInfo(errCode) . "KeyConst.wl" : Constantes standard de Windows utilisées pour les touches du clavier. Ces constantes peuvent être utilisées avec la fonction ToucheEnfoncée , dans les traitements optionnels WINDEV "Touche Enfoncée" ou "Touche relâchée" en utilisant la variable <code>_EVE.wParam</code> .	Éditeurs	X
.WLOG	Log d'exécution	Éditeurs	X
.WLR	Fichier interne de compilation	Compilation	
.WMEM	Dump mémoire	Exécution	
.WPA	Plan d'action	Éditeurs	X
.WPC	Palette de couleurs	Éditeurs	X
.WPF	Fichier des statistiques enregistré et manipulé par l' analyseur de performances .	Éditeurs	X
.WPP	Projet WINDEV Mobile	Éditeurs	X
.WPW	Fenêtre WINDEV Mobile	Éditeurs	X
.WRC	Liste des références des classes	Compilation	
.WRE	Liste des références des états	Compilation	

.WRG	Liste des références des collections de procédures	Compilation	
.WRK	Liste des références des composants	Compilation	
.WRP	Liste des références d'un projet	Compilation	
.WRR	Liste des références des requêtes	Compilation	
.WRW	Liste des références des fenêtres	Compilation	
.WSCT	Scénario de test	Exécution	X
.WSDL	Fichier de description des WebServices	Éditeurs	X
.WSM	Graphique de modélisation souple	Éditeurs	X
.WTE	Modèle d'états	Éditeurs	X
.WWF	Description d'installation WEBDEV	Éditeurs	
.WWH	Frameset , Page WEBDEV dynamique	Éditeurs	X
.WWN	Collection de procédures navigateur	Éditeurs	X
.WWP	Projet WEBDEV	Éditeurs	X
.WWT	Modèle de pages	Éditeurs	X
.WX	Configuration externe des applications WINDEV	Installation	X
.WXF	Fichier temporaire contenant la description du programme d'installation (créé par WDInst)	Installation	X
.WXRES	Résultat de test automatique	Éditeurs	
.WXT	Test automatique	Éditeurs	X
.XDD	Fichier interne à l'analyse	Analyse	X