

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 :
Analyse des besoins, prototypage et Maquettage des IHM*

Réalisé par :

M^r. FERROUDJ Nassim

M^r. KESSOURI Seddam

Jury:

Président : M^r. ALLICHE Abdennour

Examinatrice : M^{lle}. AZOUI Aicha

Promoteur : Dr. IDOUGHI Djilali

Remerciements

Au terme de notre travail, nous remercions Dieu le tout puissant qui nous a doté d'une grande volonté et d'un savoir adéquat.

Nos remerciements sont adressés également à nos chers parents pour tous sacrifices consentis à notre égard et leur énorme soutien.

Nous remercions Mr Idoughi Djilali., notre promoteur, de nous avoir aidé et encadré tout en long du projet.

Nos plus vifs remerciements s'adressent également aux membres de la commission pour avoir accepté de juger notre travail.

A tous les enseignants et les membres du département informatique de l'université ABDERRAHMANE MIRA de Bejaia.

Nous remercions aussi toute personne qui a contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

***Ferroudj Nassim:** Je dédie ce modeste travail à mes très chers parents qui m'ont vraiment soutenu pendant toutes mes études, que dieu les garde et les protège. À mon frère Adel et à ma sœur Manel, à toute la famille ferroudj, à tous mes proches sans exception. Je dédie aussi ce travail spécialement à mes chers amis et mes frères Brahim, Batiche et Elhaj Tarik, à tout le groupe de GL et à tous mes amis sans exception (Hmado, Dido, ...).*

***KESSOURI Seddam :** je dédie ce modeste travail à mes parents. À mes frères et sœurs, ma fiancée Nessrine, qui n'ont jamais cessé de m'encourager. À mes amis et à tous ceux qui ont contribué à la réussite de ce projet, spécialement je dédie ce travail à mon frère et ami Brahim.*

Table des matières

CHAPITRE 1 : Génie logiciel –Généralités	14
Introduction.....	15
1. Généralités	15
2. Cycle de vie d'un Logiciel	16
3. Les principaux modèles de cycle de vie	18
4. Qu'est-ce qu'un Atelier Génie Logiciel ?	19
Conclusion.....	21
CHAPITRE 2 : Développement multiplateforme sous Windev mobile	22
Introduction.....	23
Partie1 : Introduction à la mobilité.....	23
1. Qu'est-ce qu'une application mobile ?	23
2. Les principales plateformes mobiles	23
3. Généralités sur le développement mobile	25
4. Les outils de développement mobile	25
Partie 2 : Présentation de L'AGL Windev Mobile.....	26
1. Généralités sur Windev	26
2. Caractéristiques de Windev	26
3. Cycle de développement d'une application mobile sous Windev	27
4. Gestion de projet	28
4.1. Tableau de bord du projet	29
4.2. Structure et organisation d'un projet	30
4.3. Suivi de projet	32
4.4. Développement en équipe	33
4.5. Partage de composants	34
5. Bases de données	36
5.1. Analyse : structure d'une base de données	36
5.2. Les types de bases de données accessibles	37
5.3. Base de données HFSQL Classic	37
5.4. Base de données HFSQL Client/serveur	38
5.5. Les requêtes	39
6. WLangage	40
6.1. Définition	40
6.2. Les variables	40

6.3. Les constantes	41
6.4. Les instructions de base	41
- Les instructions de boucle :	41
6.5. Les fonctions	41
6.6. Les modes de programmation	42
7. Les Interfaces Home Machine sous Windev mobile	42
7.1 L'éditeur de fenêtres	42
7.2 Les caractéristiques principales d'une fenêtre	43
7.3 Champs d'une fenêtre	44
7.4 Gabarit d'une fenêtre	45
7.5 Les modèles de fenêtres.....	45
7.6 Les ancrages	46
7.6.1 Ancrage dans une fenêtre.....	46
7.6.2 Ancrage des champs dans une fenêtre	47
7.7 Les agencements	47
7.8 Les images.....	48
7.9 RAD/RID	49
7.9.1 RAD (Rapid Application Développement).....	49
7.9.2 RID(Rapid graphical Interface Développement)	50
Conclusion	51
CHAPITRE 3 :	52
Etude de cas- Analyse des besoins, Prototypage et maquettage des IHM d'une application mobile pour la mise en relation de services-	52
Introduction.....	53
1. Cadre de notre projet	53
2. Démarche adoptée	54
3. Description du contexte de l'application	55
5. Fonctionnalités de l'application	55
6. Planification itérative	56
7. Analyse des besoins	58
7.1 Cycle de vie d'une itération	58
7.2 Diagrammes de cas d'utilisation	58
7.2.1 Diagramme de cas d'utilisation associé au client.....	59
7.2.2 Diagramme de cas d'utilisation associé au fournisseur.....	60

7.2.3 Diagramme de cas d'utilisation associé au visiteur.....	61
7.2.4 Diagramme de cas d'utilisation global.....	62
7.3 Prototypage et maquettage de l'application.....	63
7.3.1 Le maquettage.....	63
7.3.2 Le prototypage.....	63
7.3.3. A quel moment maquette-t-on ?	64
A. La pérennité.....	65
B. L'interactivité	65
C. Le degré de fidélité	66
D. Le support.....	66
7.3.5 Importance du maquettage/prototypage dans le processus de conception d'IHM	67
7.4. Maquettes de l'application Ypiresia	69
8. Diagrammes de séquences boîte noire.....	91
8.1. Diagramme de séquences du cas d'utilisation "créer un compte"	92
8.2. Diagramme de séquences du cas d'utilisation "s'authentifier"	93
8.3. Diagramme de séquence du cas d'utilisation "rechercher un fournisseur par nom"	94
8.4. Diagramme de séquences du cas d'utilisation "rechercher un fournisseur par adresse"	95
9. Diagramme de classes d'analyse.....	102
Conclusion.....	103
CHAPITRE 4 :	104
Réalisation et évaluation.....	104
Introduction.....	105
1. Environnement matériel	105
2. La structure des fichiers de l'IHM	106
2.1 Format des fichiers de l'IHM	109
3. Capture d'écran relative aux différentes interfaces de l'application.....	110
4. Choix de l'environnement logiciel.....	117
5. Test de l'application	118
5.1 Test du projet.....	118
5.2 Test d'une seule fenêtre	118
5.3 Test d'IHM.....	119
6. Déploiement	121
Conclusion	122
<i>Conclusion générale.....</i>	<i>123</i>

<i>Perspectives</i>	124
<i>Glossaire</i>	125
<i>Références bibliographiques</i>	132
ANNEXE	134

Liste des figures

Figure1 : Les besoins du développement	15
Figure2 : Cycle de vie d'un Logiciel.....	17
Figure 3 : Modèle en cascade.....	18
Figure 4 : Modèle en V.....	18
Figure5 : Modèle itératif et incrémental.....	19
Figure 6 : Cycle de développement d'une application mobile.....	28
Figure 7 : Eléments d'une application	29
Figure 8 : Tableau de bord d'un projet.....	30
Figure 9 : Explorateur de projet	31
Figure 10 : Graphe du projet.....	31
Figure 11 : Gestion d'un projet.....	32
Figure 12 : Etapes de suivi d'un projet.....	33
Figure 13 : Exemple d'utilisation du GDS.....	34
Figure 14 : Partage d'un composant interne via le GDS	35
Figure 15 : Réutilisation d'un composant externe.....	36
Figure 16 : Manipulation d'une base de données HFSQL mobile.....	38
Figure 17 : Fonctionnement d'une base de données HFSQL Client/serveur.....	39
Figure 18 : Exemple de modèle de fenêtres.....	46
Figure 19 : Exemple d'un agencement.....	48
Figure 20 : Génération de fenêtre avec RAD ou RID.....	51
Figure 21 : Démarche adoptée.....	54
Figure 22 : Les différentes versions de notre application.....	54
Figure 23: Cycle de vie d'une itération.....	58
Figure 24 : Diagramme de cas d'utilisation associé au client.....	59
Figure 25 : Diagramme de cas d'utilisation associé au fournisseur.....	60
Figure 26 : Diagramme de cas d'utilisation associé au visiteur.....	61
Figure 27 : Diagramme de cas d'utilisation global.....	62
Figure 28 : Réévaluation par maquettage.....	63
Figure 29 : Réévaluation par prototypage.....	64
Figure 30 : Le cycle de prototypage de l'application ypiresia.....	68

Figure 31 : Maquette de la fenêtre d'authentification.....	69
Figure 32 : Maquette de la fenêtre inscription client.....	70
Figure 33 : Maquette de la fenêtre inscription personne morale.....	71
Figure 34 : Maquette de la fenêtre inscription personne physique.....	72
Figure 35 : Maquette correspondante à la rubrique paramètres du profil client.	73
Figure 36 : Maquette correspondante à la rubrique offres du profil client.....	74
Figure 37 : Maquette correspondante à la rubrique recherche du profil client...	75
Figure 38 : Maquette correspondante à la rubrique paramètres du compte fournisseur.....	76
Figure 39 : Maquette correspondante à la rubrique demandes du compte fournisseur.....	77
Figure 40 : Maquette de la fenêtre résultats de la recherche.....	78
Figure 41 : Maquette de la fenêtre détails de la recherche	79
Figure 42 : Maquette de la fenêtre modifier un compte client.....	80
Figure 43 : Maquette de la fenêtre modifier compte personne morale.....	81
Figure 44 : Maquette de la fenêtre modifier compte personne physique.....	82
Figure 45 : Maquette de la Fenêtre modifier la description.....	83
Figure 46 : Maquette de la fenêtre modifier mon adresse.....	84
Figure 47 : Maquette de la fenêtre modifier le compte Facebook.....	85
Figure 48 : Maquette de la fenêtre modifier le mot de passe.....	86
Figure 49 : Maquette de la fenêtre modifier le numéro de téléphone.....	87
Figure 50 : Maquette de la fenêtre modifier e-mail.....	88
Figure 51 : Maquette de la fenêtre modifier le pseudo.....	89
Figure 52 : Maquette de la fenêtre modifier le nom.....	90
Figure 53 : Maquette de la fenêtre modifier le prénom.....	91
Figure 54 : Diagramme de séquences boîte noire du cas d'utilisation "créer un compte"	93

Figure 55 : Diagramme de séquences boîte noire du cas d'utilisation ‘s’authentifier’	94
Figure 56 : Diagramme de séquences boîte noire du cas d'utilisation ‘rechercher un fournisseur par nom’	95
Figure 57 : Diagramme de séquences boîte noire du cas d'utilisation ‘rechercher un fournisseur par adresse’	96
Figure 58 : Diagramme de séquences boîte noire du cas d'utilisation ‘rechercher un fournisseur par service’	97
Figure 59 : Diagramme de séquences boîte noire du cas d'utilisation ‘visualiser les informations d’un fournisseur’	98
Figure 60 : Diagramme de séquences boîte noire du cas d'utilisation ‘afficher les détails sur un fournisseur’	99
Figure 61 : Diagramme de séquences boîte noire du cas d'utilisation ‘modifier le compte’	100
Figure 62 : Diagramme de séquences boîte noire du cas d'utilisation ‘supprimer le compte’	101
Figure 63 : Diagramme de séquence boîte noire du cas d'utilisation ‘se déconnecter’	102
Figure 64 : Diagramme de classes du domaine	103
Figure 65 : Structure des fenêtres simples	107
Figure 66 : Structure des fenêtres internes	108
Figure 67 : Structure des modèles de fenêtres	109
Fenêtre 68 : Structure des modèles de champs	110
Fenêtre 69 : Structure des fichiers de test	110
Figure 70 : Fenêtre d’authentification en mode portrait	111
Figure 71 : Fenêtre d’authentification en mode paysage	112
Figure 72 : Fenêtre d’inscription client en mode portrait	113
Figure 73 : Fenêtre d’inscription client en mode paysage	113
Figure 74 : Fenêtre d’inscription personne physique en mode portrait	114
Figure 75 : Fenêtre d’inscription personne physique en le mode paysage	114
Figure 76 : Fenêtre Modification d’informations personnelles en mode portrait	115
Figure 77 : Fenêtre Modification d’informations personnelles en mode paysage	115
Figure 78 : Fenêtre Modifier la photo de profil d’un client en mode portrait	116
Figure 79 : Fenêtre Modifier la photo de profil d’un client en mode paysage	116

Figure 80 : Fenêtre correspondante à la rubrique Paramètres du profil client en mode portrait	117
Figure 81 : Fenêtre correspondante à la rubrique Paramètres du profil client en mode paysage.....	117
Figure 82 : Fenêtre Détails Résultats de la recherche	118
Figure 83 : Fenêtre correspondante à la rubrique Recherche du profil client en mode portrait.....	119
Figure 84 : Agencement de la fenêtre Modifier la photo de profil d'un client.	122
Figure 85 : Ancrage de la fenêtre d'authentification d'un client.....	122
Figure 86 : Génération de l'application.....	123

Liste des tableaux

Tableau 1 : Tableau de répartition des taches.....	45
Tableau 2 : Tableau des exigences fonctionnelles.....	48
Tableau 3 : Tableau des itérations.....	49
Tableau 4 : Tableau des extensions de l'IHM.....	111

Introduction

générale

La recherche d'un médecin, d'un artisan, d'un magasin, ou simplement un fournisseur de service, s'avère souvent une tâche difficile et coûteuse en temps. Le consommateur qui veut le service le plus proche, le mieux adapté à ses attentes et le fournisseur qui veut être le plus proche du consommateur.

L'informatique cherche sans cesse à améliorer notre quotidien et adapter le consommateur à un monde en évolution permanente. Le consommateur et le fournisseur de service veulent une communication souple et rapide, en intégrant les nouvelles technologies de l'information et de la communication.

D'autre part, La principale avancée des TIC dans les quinze dernières années réside dans la mobilité. L'utilisation de la téléphonie mobile s'est bien amplifiée et son développement s'est vu accroître dans le monde entier. En 2010, la moitié des habitants ont accès à cette technologie. Les Smartphones basées sur les systèmes d'exploitation propriétaires et libres s'intègrent dans notre paysage et dans nos usages d'une manière croissante pour devenir un accessoire qui nous ouvre une porte vers de nouveaux services.

Dans ce cadre, L'intérêt typique de ce projet est de développer une application mobile pour mettre en relation le consommateur et le fournisseur de service.

Ainsi, nous articulons notre rapport autour de quatre chapitres :

Le premier et deuxième chapitre portent successivement sur le génie logiciel et le développement mobile multiplateformes sous Windev.

Le troisième chapitre est consacré pour l'analyse des besoins, prototypage et maquettage des IHM détaillé de l'application.

Le quatrième chapitre aborde la réalisation finale et le processus de test et d'évaluation.

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 d'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 coût 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]

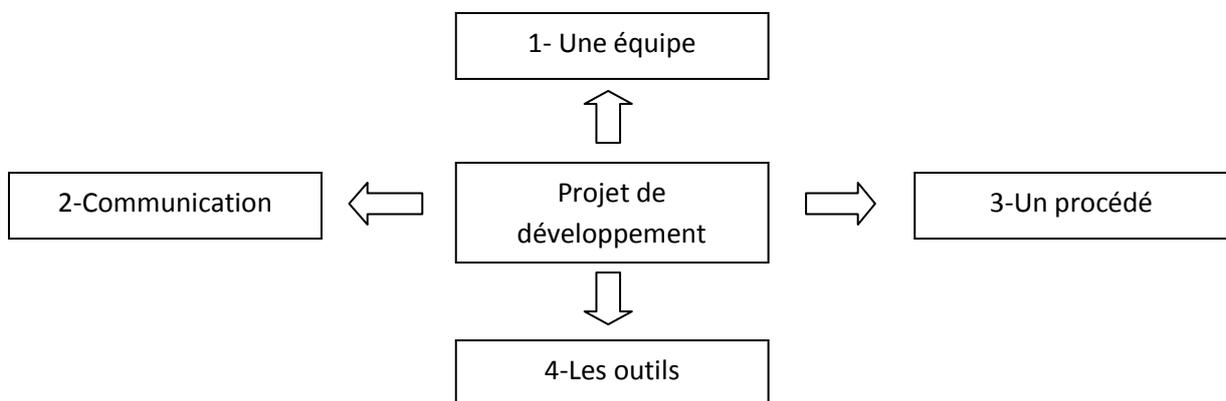


Figure1 : Les besoins du développement

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.

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.

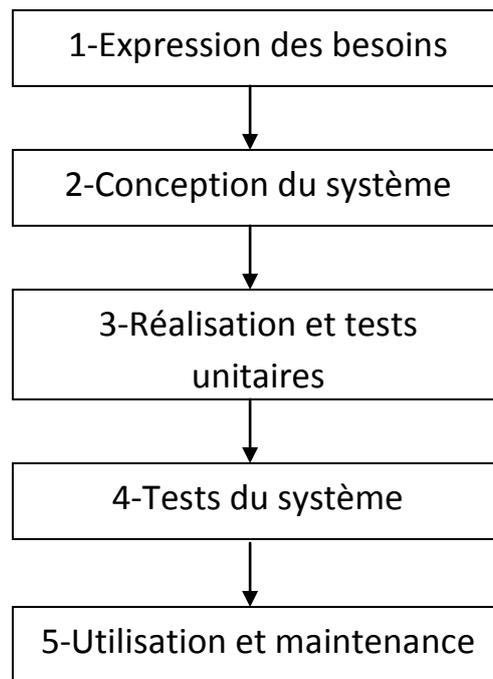


Figure2 : Cycle de vie d'un Logiciel

- 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 autre

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]

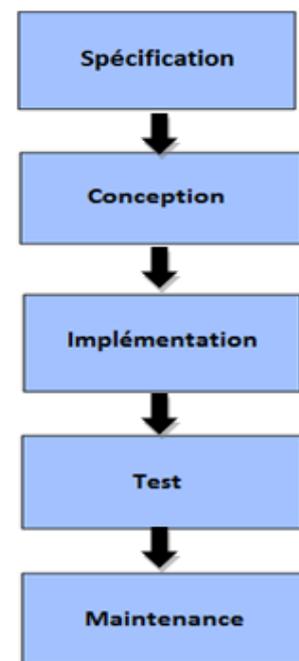


Figure 3 : Modèle en cascade

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]

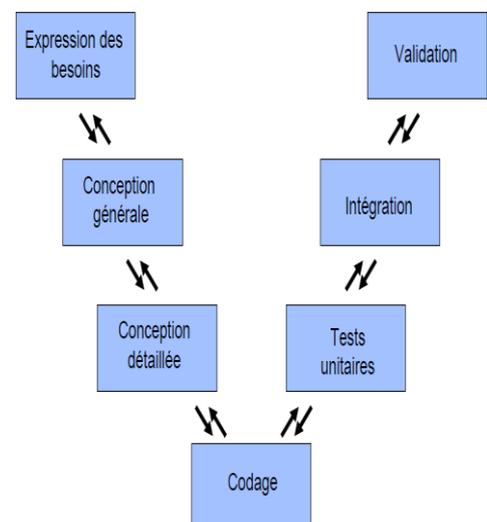


Figure 4 : Modèle en V

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]

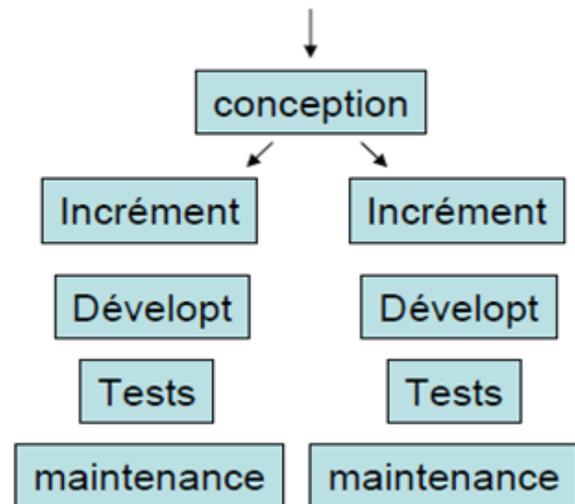


Figure5 : Modèle itératif et incrémental

4. Qu'est-ce qu'un Atelier 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:

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 (TRAMIS View), un générateur de prototypes (TRAMIS Dialog), ...

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 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 (l'environnement Loops fonctionne sur une machine dédiée à Loops), Oz ... 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]

3- 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 any where" 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 un logiciel de type application mobile dans l'environnement multiplateforme Windev Mobile, que nous décrivons dans le chapitre 2.

CHAPITRE 2 :

Développement

multiplateforme

sous Windev

mobile

Introduction

Dans ce chapitre nous décrirons dans la première partie l'environnement mobile, ces principales plateformes et ces outils de développement d'une manière générale. Et dans la deuxième partie nous allons présenter une description de l'environnement de développement Windev Mobile, son architecture, ses fonctionnalités et nous allons détailler bien le principe d'IHM sous Windev Mobile.

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.

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.

BlackBerryOS: 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 : créé 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 de 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 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 CocoaTouch.

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

WinDev est un AGL (Atelier de Génie Logiciel). Il permet de développer des applications dans tous les domaines : Gestion - Industrie - Médical - MultiMedia - Internet - Accès distant ...

Il est un outil de développement complet qui intègre tous les outils nécessaires au cycle de réalisation d'une application.

Windev offre un langage de cinquième Génération, le WLangage, qui est très simple et très puissant, selon son éditeur une semaine suffit pour maîtriser toute sa puissance. 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.[9]

Windev est multi cibles, il est édité sous trois éditions :

- Windev : destiné pour le développement d'applications bureau.
- WebDev : destiné pour le développement de sites et d'applications web.
- Windev mobile : destiné pour le développement d'applications mobiles.

2. Caractéristiques de Windev

- Le développeur se concentre sur les besoins métier, et pas sur le code: les applications répondent aux besoins
- les délais de développement et d'évolution sont réduits.
- Code automatiquement managé et sécurisé
- Environnement totalement intégré, pour une industrialisation sans contraintes et un développement agile.

- Il offre un générateur RAD d'applications complètes, permettant de créer ses propres modèles (patterns).
- Il offre un langage de programmation puissant de 5^{ème} génération : le WLangage.
- Développement multiplateforme et portabilité des applications : un seul code pour plusieurs cibles.
- Richesse fonctionnelle et rapidité d'accès aux données.
- ouverture à toutes les Bases de Données: AS/400, Oracle, SQL Server, DB2, Informix, MySQL, PostgreSQL, SQLite, Access, Progress, Sybase.
- accès aux API
- Support PDF, Java, PHP, SOAP, XML, Word, Excel, JQuery, JSON.
- création facile d'interfaces de qualité (nombreux gabarits livrés).
- Auto-formation rapide pour un informaticien.[10]
-

3. Cycle de développement d'une application mobile sous Windev

WinDev propose un cycle de vie en quatre phases : Conception, développement, test et génération et déploiement, comme illustré dans la figure 6.

Conception : Il est possible de concevoir une application à partir d'un simple cahier des charges, d'une modélisation UML des traitements ou même à partir de fichiers de données préexistants.

Développement : La création du projet et de l'analyse est réalisée à l'aide d'assistants très complets. Le développement peut être effectué en mode RAD (Rapid Development Application) avec génération automatique du code et des IHM ou être le résultat d'une création manuelle des différents éléments du projet.

Test et génération : WINDEV Mobile offre toute une panoplie d'outils de tests automatiques pour garantir la fiabilité des applications et assurer la non-régression entre les phases de développement.

Déploiement : Le déploiement d'une application WINDEV Mobile peut se faire de nombreuses façons : directement sur le Mobile, par téléchargement, via Play Store, MarketPlace, App Store ou Windows Store.[11]

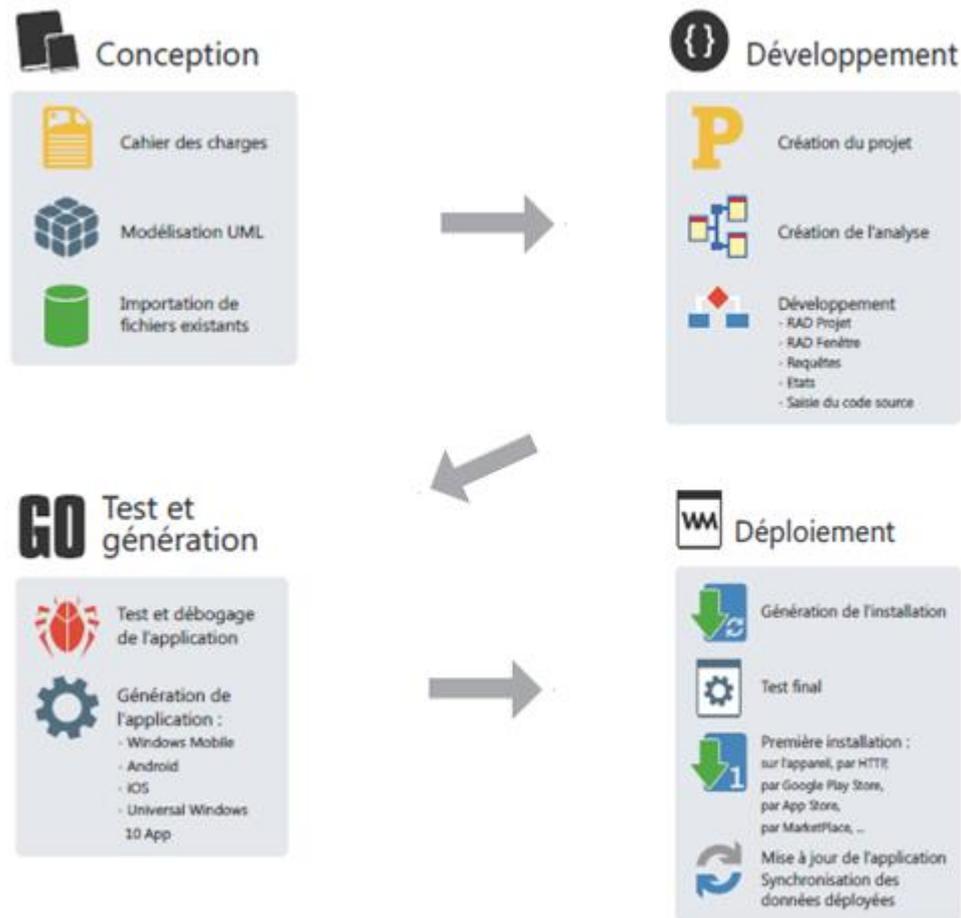


Figure 6 : Cycle de développement d'une application mobile

4. Gestion de projet

Le développement d'une Application mobile avec WINDEV Mobile repose sur deux éléments principaux :

- 1- le Projet
- 2- l'Analyse.

1- Un Projet WINDEV Mobile est un ensemble d'éléments : fenêtres, champs, classes, composants, ... dont l'assemblage permet de réaliser une application.

2- Une Analyse WINDEV Mobile regroupe la description des fichiers de données de l'application.[11]

Une application est construite à partir d'un projet associé à une analyse (Figure 7).

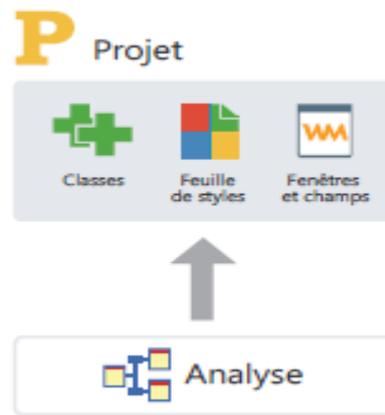


Figure 7 : éléments d'une application

4.1. Tableau de bord du projet

Le tableau de bord du projet est un élément indispensable à la gestion de projets WINDEV Mobile.

Le tableau de bord permet d'avoir une vision globale et synthétique de l'état d'avancement d'un projet.

Le tableau de bord du projet est composé de différents indicateurs sur le contenu du projet :

- statistiques sur le projet
- incidents
- tâches
- état des tests automatiques
- résultat des différents audits
- liste des éléments extraits du GDS (Gestionnaire de Sources). [11]
- résultat des plans d'action (intégration continue)

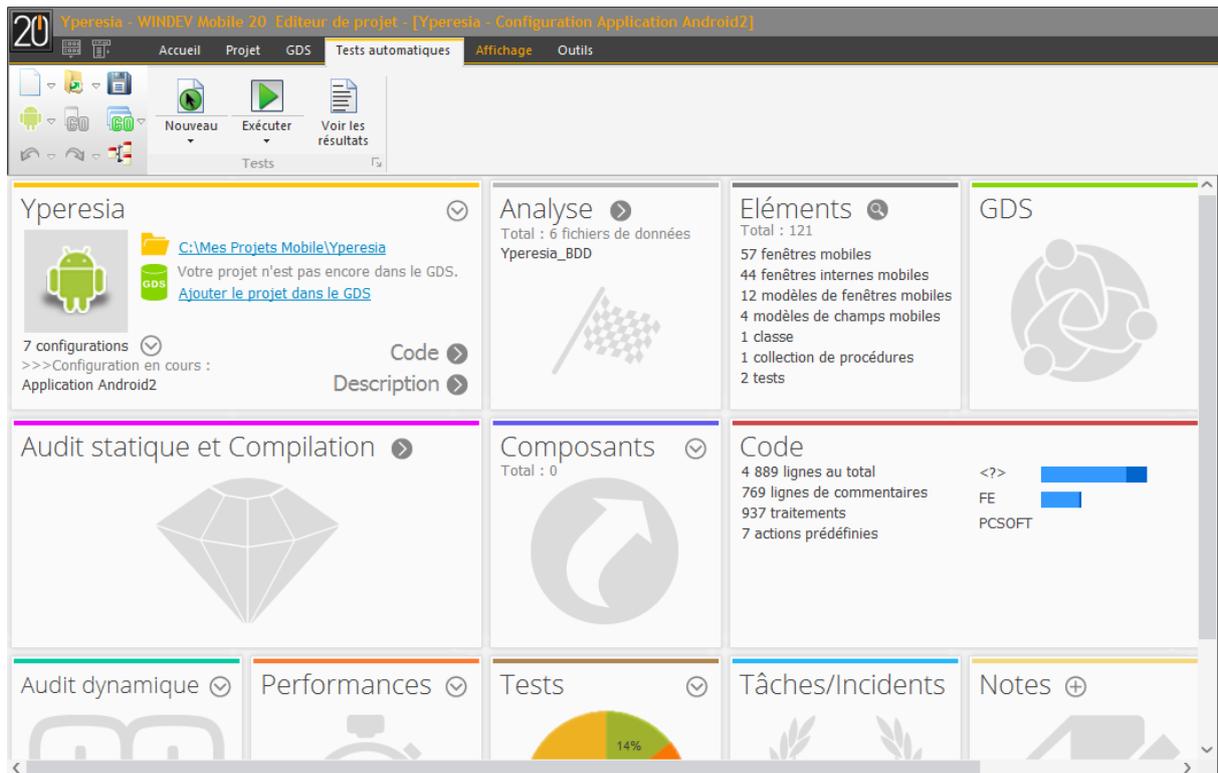


Figure 8 : Tableau de bord d'un projet

Les différents éléments de ce tableau de bord sont présentés sous forme de Widgets. Ces Widgets peuvent être paramétrés, déplacés, activés, désactivés et il est possible d'ajouter de nouveaux indicateurs.

La configuration du tableau de bord est sauvegardée par utilisateur. [11]

4.2. Structure et organisation d'un projet

Un projet est composé de plusieurs éléments : Fenêtres, états, requêtes, classes, collections de procédures, feuilles de styles, rubriques et composants internes et externes.

Les éléments qui composent un projet sont tous listés dans le volet "Explorateur de Projet". Par défaut, les éléments sont rangés en fonction de leur type : fenêtres, états, classes...

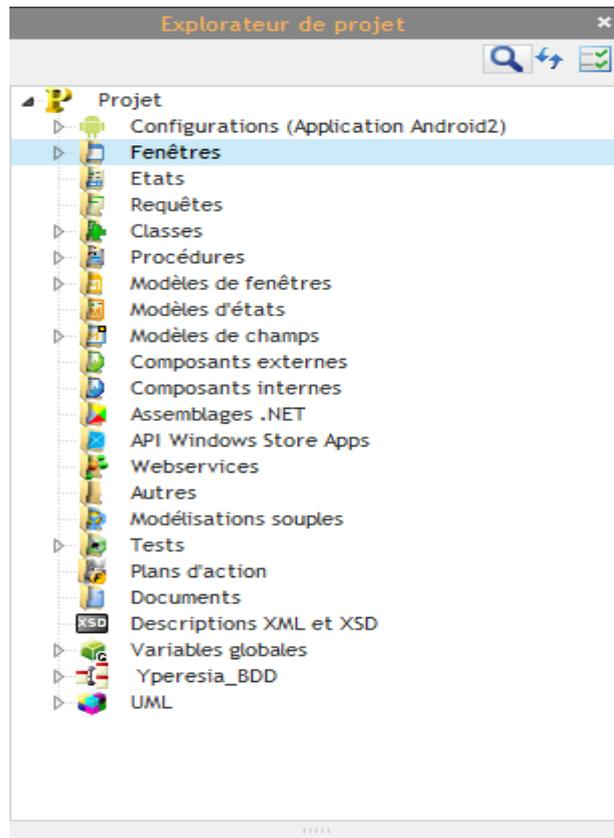


Figure 9 : Explorateur de projet

Dans les projets de taille importante, il est souvent plus pertinent de regrouper les éléments correspondant à la même fonctionnalité. Pour cela, il suffit de créer des "perso-dossiers" dans l'arborescence de l'explorateur de projet et de glisser les différents éléments dans ces dossiers et des éléments peuvent être communs à plusieurs perso-dossiers. Il est ainsi plus simple de travailler sur une partie de l'application. [11]

Il est aussi possible de visualiser la structure du projet avec le graphe.

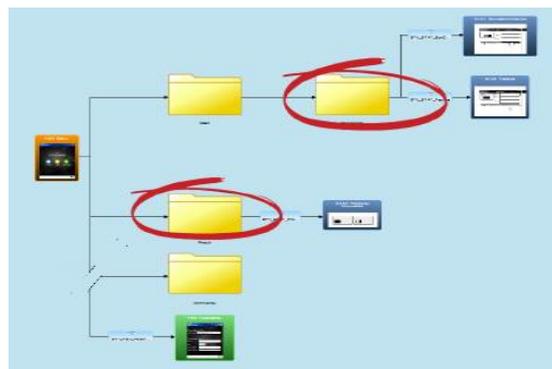


Figure 10 : Graphe du projet

4.3. Suivi de projet

Pour optimiser la gestion d'un projet de développement, WINDEV Mobile propose d'utiliser le Centre de Suivi de Projets.

Le Centre de Suivi de Projets permet de :

- Gérer les exigences d'un projet
- Gérer le suivi d'un projet (planning des tâches à réaliser)
- Gérer les bugs et les évolutions signalés sur un projet.

Le Centre de Suivi de Projets permet de gérer un projet de développement. Il suffit pour cela de :

- définir les différents intervenants du projet.
- définir les exigences (avec les différents éléments qui leur sont associés).

Chaque développeur effectue les différentes tâches qui lui sont affectées et on peut à tout moment suivre l'état d'avancement du projet.

On définit d'abord les exigences qui correspondent aux cas d'utilisation et on établit un planning des tâches. Une tâche peut correspondre à un élément du projet comme une fenêtre ou un état...

A partir des tâches on définit les règles métier qui permettent de définir des modes opératoires précis ou de préciser un traitement particulier. [11]



Figure 11 : Gestion d'un projet

Après avoir listé l'ensemble des tâches d'un projet, le Centre de Suivi de Projets s'occupe de tout. La saisie du temps passé est quasi automatique.

Lors de l'ouverture du projet concerné, le Centre de Suivi de Projets demande ou indique la tâche en cours. Dès qu'une tâche est réalisée, il suffit d'indiquer que cette tâche est terminée et de spécifier la nouvelle tâche.

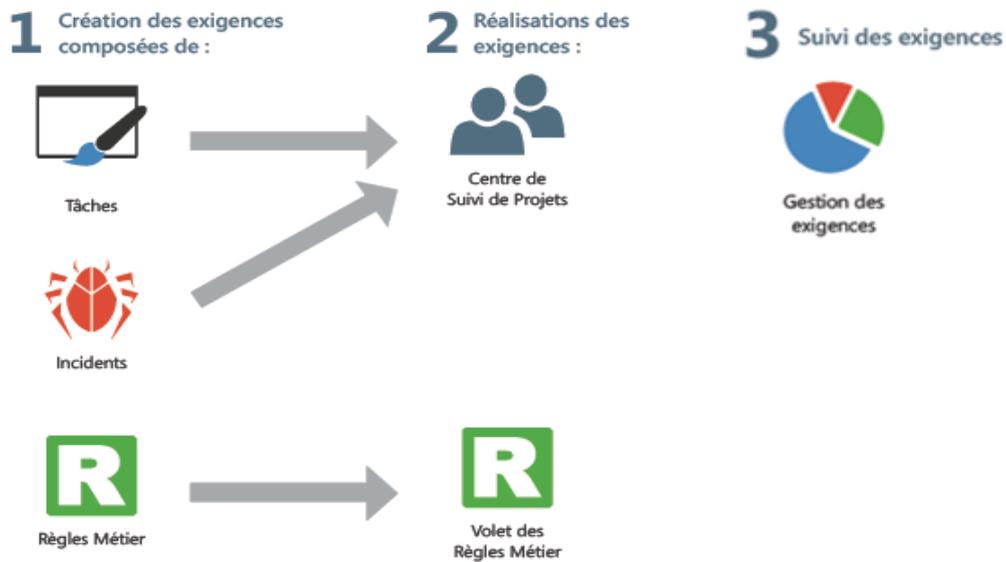


Figure 12 : étapes de suivi d'un projet

4.4. Développement en équipe

Pour simplifier le développement en équipe, WINDEV Mobile propose un gestionnaire de sources GDS. Ce gestionnaire de sources permet à plusieurs développeurs de travailler simultanément sur le même projet et de partager des éléments entre différents projets.

Une base appelée base de sources regroupe tous les éléments du projet et chaque poste possède une copie en local des éléments nécessaires au développement.

Il est possible de partager les éléments présents dans le GDS :

- via un réseau
- via Internet
- via le Cloud
- en mode déconnecté. Dans ce cas, les éléments sur lesquels un travail spécifique devra être effectué seront extraits du GDS par exemple lors de la connexion du portable au système général.

La base de sources garde des versions du projet et à chaque fois que le projet est modifié par un développeur une nouvelle version est créée et enregistrée. Si un élément du projet est extrait il ne pourra pas être extrait une seconde fois.[11]

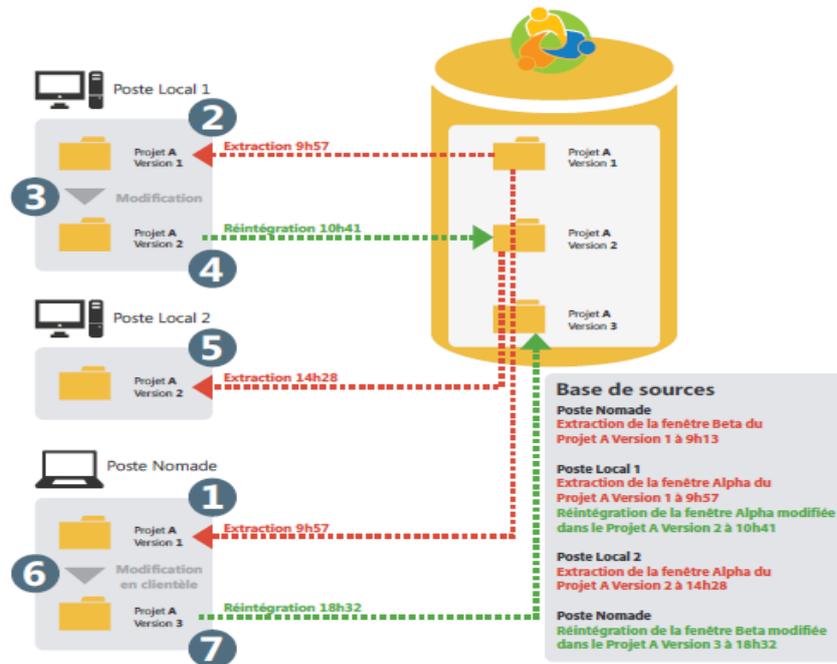


Figure 13 : exemple d'utilisation du GDS

4.5. Partage de composants

Les composants logiciels sont des éléments essentiels pour la réutilisabilité. Windev propose deux types de composants : internes et externes.

- Les composants internes :

Un composant interne est un regroupement d'éléments d'un projet. Ce regroupement permet de :

- Organiser un projet : créer des composants internes pour regrouper les éléments d'un projet, par exemple par fonctionnalité.
- Partager des éléments entre différents projets, par l'intermédiaire du GDS.

Un composant est enregistré dans un fichier WCI.[11]

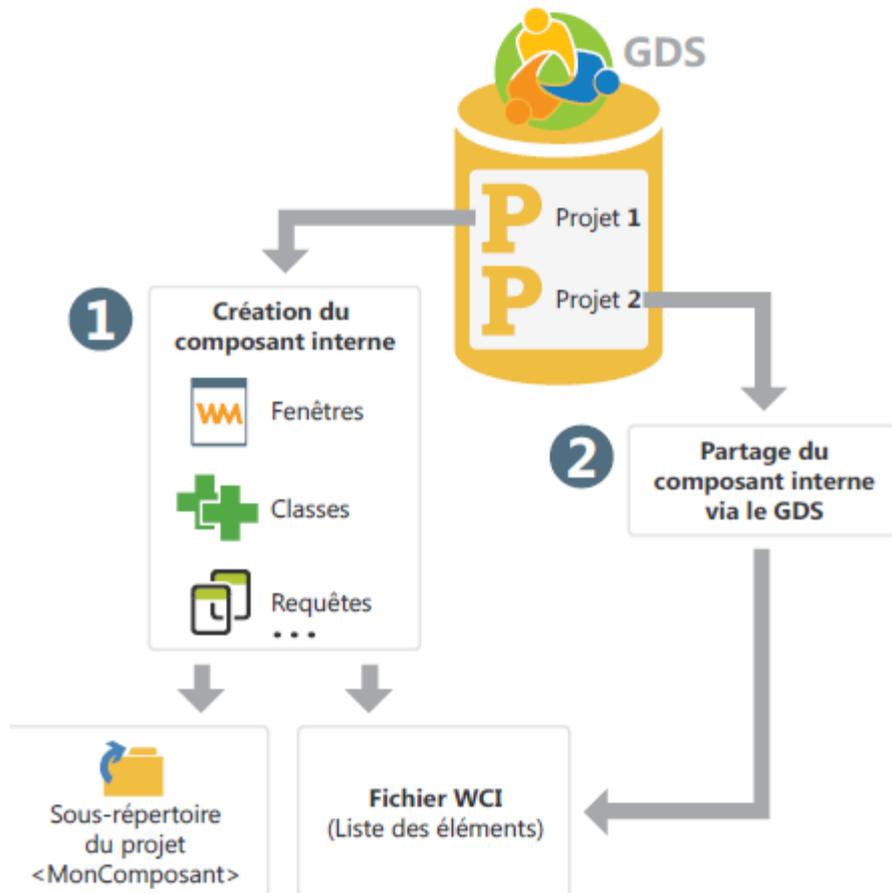


Figure 14 : Partage d'un composant interne via le GDS

- Les composants externes :

Un composant externe est un ensemble d'éléments WINDEV Mobile : fenêtres, états, analyse, ... Cet ensemble d'éléments effectue une fonctionnalité précise. Par exemple, un composant externe peut correspondre à une des fonctionnalités suivantes :

- Envoi de SMS
- Envoi d'emails
- ...

Un composant externe peut être redistribué à d'autres développeurs. Ces développeurs pourront ainsi intégrer simplement la fonctionnalité proposée par le composant externe dans leur application. Le composant externe sera donc intégré à l'application et distribué avec l'application.[11]

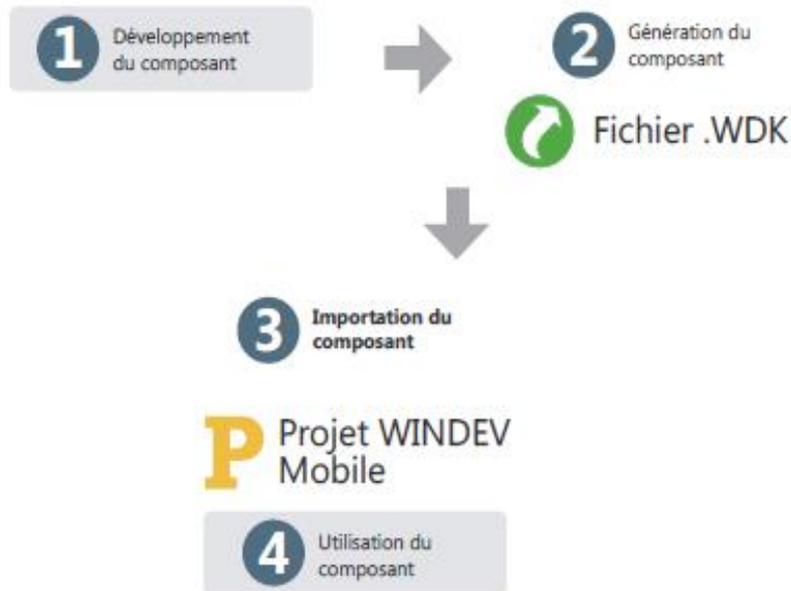


Figure 15 : réutilisation d'un composant externe

5. Bases de données

5.1. Analyse : structure d'une base de données

Lorsqu'un projet WINDEV, WEBDEV ou WINDEV Mobile utilise des fichiers de données, ce projet doit être associé à une analyse.

Une analyse permet de décrire les structures de données (fichiers de données, rubriques, ...) utilisées dans le projet.

L'éditeur d'analyses permet de créer très simplement une analyse et l'enregistrer sous un fichier d'extension WDA.

L'analyse d'un projet WINDEV Mobile correspond au MLD (Modèle Logique des données). Toute la structure et l'organisation des données sont décrites : les données sont regroupées par fichier. Chaque fichier contient plusieurs données appelées rubrique.

Dans l'analyse, il est possible de lier la description d'un fichier de données à un type de fichier (HFSQL Mobile, HFSQL Client/serveur, SQLite, ...).

5.2. Les types de bases de données accessibles

Windev mobile permet à une application l'accès à plusieurs types de données en d'autre terme elle couvre la totalité des bases de données existantes.

Les bases de données accessibles sont :

- Oracle
- SQL Server
- MySQL
- Sybase
- Informix
- DB2
- AS/400
- Progress
- SQLite
- PostgreSQL
- Access
- ODBC

5.3. Base de données HFSQL Classic

Le format HFSQL Classic (également appelé HFSQL Mobile) est le format de base de données fourni avec WINDEV Mobile. Ce format de base de données est compatible entre WINDEV, WINDEV Mobile et WEBDEV.

Il s'agit d'un SGBD Relationnel redistribuable gratuitement.

Ce format est utilisable sur les plateformes Windows Mobile, iPhone, iPad, Android et Universal Windows 10 App. Ce format est identique au format HFSQL Classic de WINDEV standard et de WEBDEV (fichier ".WDD", fichiers de données, ...). 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".

Chaque fichier de données peut être manipulé par une application WINDEV Mobile. Ces manipulations sont réalisées grâce aux fonctions HFSQL (fonctions *Hxxx*). [11]

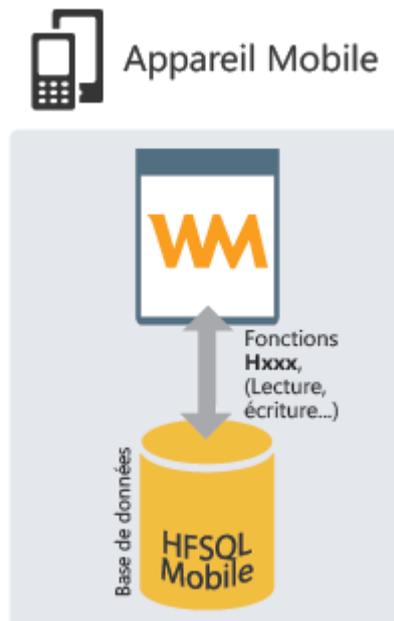


Figure 16 : Manipulation d'une base de données HFSQL mobile

5.4. Base de données HFSQL Client/serveur

Une application WINDEV Mobile HFSQL peut également fonctionner en mode Client/ Serveur.

Les caractéristiques du mode Client/serveur sont les suivantes :

- Une application HFSQL Client/serveur est exécutée sur différents périphériques mobiles (appelés machines clientes).
- Les fichiers de données sont présents sur un poste serveur. Seul le poste serveur accède physiquement aux fichiers de données.
- L'ensemble des traitements (requête, lecture/ajout dans un fichier de données, ...) est réalisé sur le serveur. [11]

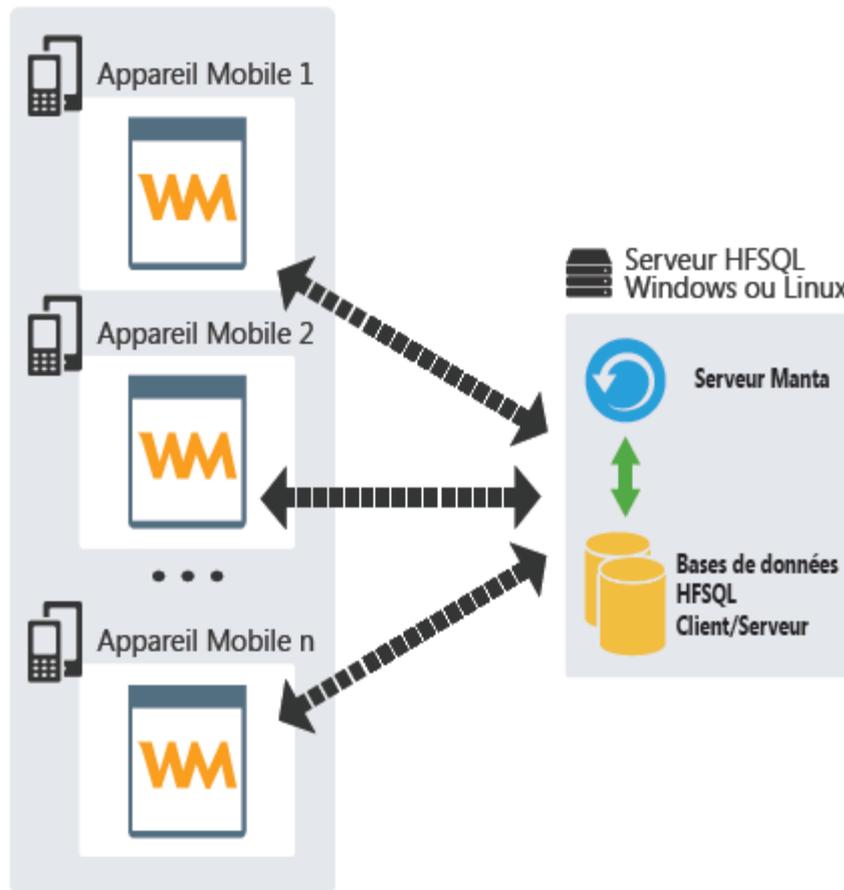


Figure 17 : Fonctionnement d'une base de données HFSQL Client/serveur

5.5. Les requêtes

Une requête sert à interroger une base de données pour visualiser, insérer, modifier ou supprimer des données. Sa structure définit les données manipulées. Elle peut interroger un ou plusieurs fichiers de données.

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

En programmation, une requête peut être manipulée de la même façon qu'un fichier de données. En particulier, elle peut être associée à un champ d'affichage (un champ Table par exemple) qui présentera les données retournées par la requête. [11]

6. WLangage

6.1. 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. [11]

6.2. 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

6.3. Les constantes

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

6.4. 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. [11]

6.5. 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.

6.6. 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. [11]

7. Les Interfaces Home Machine sous Windev mobile

WinDev mobile possède un éditeur de fenêtres particulièrement évolué permettant de réaliser facilement et rapidement tous les types d'IHM possibles.

De nombreuses fonctionnalités permettent d'obtenir simplement des applications intuitives et agréables d'utilisation :

- des champs puissants et variés.
- un mécanisme d'ancrages permettant à l'IHM de s'adapter automatiquement à la taille de l'affichage.
- un système de compilation de l'IHM avec détection des erreurs (titres vides, libellés non traduits, chevauchements, ...).

7.1 L'éditeur de fenêtres

Les fenêtres constituent la base de l'IHM (Interface Homme Machine) d'une application. WinDev Mobile permet de créer des interfaces personnalisées grâce à l'éditeur de fenêtres.

L'éditeur de fenêtres permet de :

- créer les fenêtres de l'application mobile.
- créer des champs dans les fenêtres. Grâce à ces champs, les utilisateurs pourront visualiser et saisir les informations nécessaires à l'application. Pour chaque type de champs, de nombreuses caractéristiques sont paramétrables aussi bien sous l'éditeur que par programmation.

Donc une fenêtre a pour but d'afficher, de consulter et de saisir des informations. Ces informations peuvent provenir de fichiers de données d'une analyse, de fichiers externes ou de requêtes.

WinDev mobile propose plusieurs solutions pour créer les fenêtres d'un projet :

- **Créer une fenêtre à partir de la description de l'analyse.**
- **Créer des fenêtres vierges standard** : la fenêtre est créée sans aucun champ.
- **Créer des fenêtres basées sur un modèle** : Ces fenêtres sont créées à partir du modèle de fenêtres sélectionné.
- **fenêtres importées** : Ce choix permet grâce à un assistant d'importer l'IHM d'une fenêtre non WinDev mobile.
- **Fenêtres créées par le RAD** : Ces fenêtres sont créées par le module RAD de WinDev mobile à partir de la description de la base de données. Ces fenêtres contiennent tout le code nécessaire à leur fonctionnement.
- **Fenêtres créées par le RID** : Ces fenêtres sont créées par le module RID de WinDev mobile à partir de la description de la base de données. La programmation de la fenêtre reste à la charge du développeur.
- **Créer des fenêtres internes** : Ces fenêtres peuvent être utilisées dans des champs de type "fenêtre interne" à plusieurs endroits du projet ou dans des projets multiples.

Dans tous les cas, les fenêtres peuvent être modifiées après leur création.

7.2 Les caractéristiques principales d'une fenêtre

Une fenêtre peut accueillir tous les types de champs disponibles. Une fenêtre peut être :

- **être redimensionnable** : l'utilisateur peut agrandir ou diminuer la taille de la fenêtre. Selon les caractéristiques d'ancrage définies pour chacun des champs, la position des champs dans la fenêtre pourra varier lors de ce redimensionnement.
- **avoir une image de fond** : cette image est affichée en fond de la fenêtre et les champs viennent se superposer à cette image. Il est également possible de définir des caractéristiques de redimensionnement grâce au mode 9 images.
- **être détournée** : la fenêtre aura ainsi une forme originale (fenêtre en forme de cercle, d'ordinateur, ...).
- **avoir un menu déroulant** : ce menu permet aux utilisateurs d'accéder plus rapidement aux principales fonctionnalités de l'application

- **avoir une barre de messages** : cette barre de messages affichera les messages d'aide associés aux différents champs de la fenêtre. Cette barre de message permet également d'afficher des informations supplémentaires.
- **être multilingue** : toutes les langues définies pour la fenêtre seront disponibles pour tous les champs de la fenêtre. Une fenêtre peut être associée à plus de langues que le projet (cas par exemple des fenêtres partagées entre plusieurs projets).

7.3 Champs d'une fenêtre

Windev mobile propose tous les types des champs nécessaires aux développements des applications mobile, il dispose de plusieurs champs peuvent être manipulés dans une fenêtre :

- Champ de saisie.
- Libellé.
- Bouton.
- Image.
- Sélecteur.
- Interrupteur.
- Liste.
- Combo.
- Liste image
- Table.
- Zone répétée.
- Table hiérarchique.
- Arbre.
- Jauge.
- Ascenseur.
- Potentiomètre.
- Spin.
- Onglet.
- Champ ActiveX.
- Forme.
- Champ HTML.
- Bouton état sur table. Ce bouton permet de créer un état affichant l'ensemble des données présentes dans une table d'une fenêtre.
- Superchamp (champ de type avancé).
- Fenêtre interne.
- Champ Séparateur.
- Champ ".NET 2.0 (WinForm)".
- Modèle de champs.
- Champ Web Caméra.
- Calendrier.
- Boîte à outils.
- Champ Multimédia.

7.4 Gabarit d'une fenêtre

Les gabarits permettent de créer rapidement des interfaces professionnelles. Un gabarit définit le look de la fenêtre mais également le look de tous les champs qui seront utilisés dans cette fenêtre. Ainsi aucune faute de goût n'est possible.

Pour uniformiser l'IHM d'un projet, WinDev mobile propose plusieurs interfaces prédéfinies. Ces interfaces sont appelées des gabarits.

En appliquant un gabarit à un projet, l'ensemble des fenêtres et des états du projet adopteront la même IHM. Il est possible de créer ses propres gabarits.

7.5 Les modèles de fenêtres

Présentation :

WinDev mobile permet de créer des modèles de fenêtres. Ces modèles contiennent tous les éléments graphiques communs à l'ensemble des fenêtres de l'application.

Les modifications effectuées dans un modèle de fenêtres sont automatiquement reportées sur toutes les fenêtres utilisant ce modèle. Si certaines fenêtres ne doivent plus évoluer selon le modèle, il est possible de les désolidariser du modèle. Il est également possible de désolidariser uniquement certains champs provenant du modèle ou certains paramètres de ces champs (c'est le concept d'héritage).

Héritage et surcharge d'IHM et de code :

L'héritage est disponible au niveau des modèles de fenêtres aussi bien pour le code que pour l'IHM.

Pour chaque fenêtre où le champ est utilisé, il est possible de dériver le champ du modèle en désolidarisant un ou plusieurs paramètres (par exemple, désolidariser la position du champ, tout en conservant les autres évolutions sur le champ comme le code ou le style). Il est possible par exemple d'hériter des caractéristiques d'un ensemble de champs, mais de changer le positionnement des champs et de surcharger le code d'un autre champ du modèle.

Un modèle de fenêtres facilite le respect de la charte graphique d'une application. [11]



Figure 18 : exemple de modèle de fenêtres

7.6 Les ancrages

L'ancrage est un mécanisme puissant pour le responsive design il permet de gérer simplement le changement d'orientation d'une application Mobile ou le passage d'une application pour téléphone à une tablette.

7.6.1 Ancrage dans une fenêtre

Dans une fenêtre, les ancrages définissent la manière dont le contenu d'une fenêtre va se comporter lorsque la fenêtre est agrandie par l'utilisateur.

Par exemple, un champ peut être ancré :

- en position, et donc se déplacer lors du redimensionnement du navigateur.

- en taille, et donc s'agrandir en largeur ou hauteur lors du redimensionnement du navigateur.

7.6.2 Ancrage des champs dans une fenêtre

Les fenêtres peuvent être redimensionnables à l'exécution d'une application WinDev mobile. Grâce au mécanisme d'ancrage, la taille et la position des champs s'adaptent automatiquement lors d'un redimensionnement de la fenêtre.

La gestion de l'ancrage d'un champ est constituée de deux paramètres :

- **L'ancrage du champ** : ce paramètre permet de définir la modification à appliquer au champ en fonction du changement de la taille de la fenêtre. Le champ peut se déplacer vers la gauche et ou vers le bas, s'agrandir en largeur et/ou en hauteur, etc.
- **Le taux d'ancrage** : ce paramètre permet de gérer le pourcentage de déplacement ou d'étirement du champ. Grâce à ce paramètre, il est possible par exemple de :
 - laisser un champ centré en largeur dans la fenêtre quel que soit sa taille,
 - agrandir proportionnellement certains champs de la fenêtre.

Ces paramètres peuvent être configurés :

- Sous l'éditeur de fenêtres.
- Par programmation grâce aux propriétés WLangage.

7.7 Les agencements

Les agencements permettent, en quelques clics, de définir plusieurs mises en pages pour les IHM mobiles.

A l'exécution, Windev mobile utilise automatiquement l'agencement le plus adapté aux périphériques d'exécution, en fonction de la résolution, de l'orientation et du système d'exploitation.

L'agencement permet de définir plusieurs vues d'une fenêtre dans un même projet sans dupliquer cette fenêtre.

Les agencements permettent de gérer très facilement :

- Les différentes résolutions : tablettes, téléphones.
- Les différences entre les systèmes d'exploitation : Android / iOS / Windows phone / Windows store apps.
- Les différences entre les modes : portrait et paysage.[11]

- Les agencements sont numérotés dans leur ordre de création en édition.
- Si le terminal change d'orientation ou si la fonction FenChangeAgencement est exécutée, l'agencement change

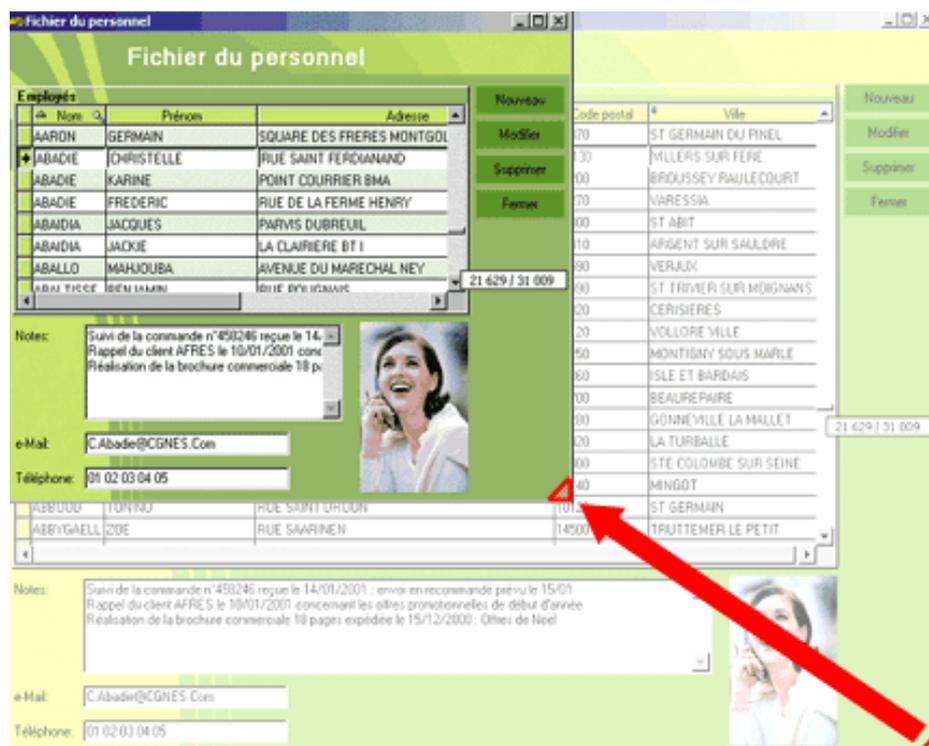


Figure 19 : Exemple d'un agencement

7.8 Les images

Windev mobile permet de gérer les images avec une puissance et une facilité étonnante.

La gestion de tous les formats est intégrée en standard, sans avoir besoin de coder quoi que ce soit.

Une image peut être stockée soit en tant que fichier indépendant, soit directement dans une ligne d'une table de données HFSQL.

Windev permet de capturer dynamiquement une image affichée, cette image pourra être sauvegardée dans un fichier (BMP, JPEG, PNG,..).

Les formats d'image qui gère Windev mobile sont comme suite :

- BMP
- RLE
- WMF
- ICO
- SVG
- PCX
- JPEG
- TIF
- ION
- GIF
- Animé
- PCD
- PSD
- PNG
- TGA
- PDF

Windev mobile permet aussi de rajouter des effets sur les images : défilement, apparition,... Ces effets se définissent sous l'éditeur ou par programmation.

7.9 RAD/RID

Le RAD (Rapid Application Development) et le RID (Rapid graphical Interface Design) permettent de créer des fenêtres à partir :

- de l'analyse liée au projet.
- des patterns RAD standard ou personnalisés.
- des gabarits.

7.9.1 RAD (Rapid Application Développement)

Le RAD (Rapid Application Development) permet de créer rapidement des fenêtres à partir de modèles prédéfinis. Par défaut, 10 modèles de fenêtre RAD sont proposés par WinDev mobile.

Chaque modèle correspond à un fichier Script, d'extension .MDL. Ce fichier comprend :

- La description du modèle RAD, affiché lors du lancement de l'assistant de création de fenêtres
- Les différentes étapes de l'assistant
- La description de la fenêtre générée et son code
- La description des champs générés et leur code

En génération RAD, les fenêtres générées contiennent tout le code nécessaire à leur fonctionnement. Ces fenêtres peuvent être testées immédiatement, avec les données présentes sur le poste de développement.

Il est possible de modifier un modèle existant ou d'ajouter un nouveau modèle grâce à l'éditeur de pattern(ou modèle RAD personnalisé) et qui permet aussi de simplifier la création des fichiers MLD et MDE utilisé par les modèles du RAD.

Il est possible de :

- créer un nouveau pattern
- modifier un pattern existant

Les modèles RAD fournis en standard peuvent ainsi être personnalisés.

7.9.2 RID(Rapid graphical Interface Développement)

Le RID (Rapid graphical Interface Design) permet de créer des fenêtres à partir :

- de l'analyse liée au projet.
- du Pattern RID sélectionné.
- des gabarits.

Les fenêtres générées contiennent uniquement les champs liés aux rubriques de l'analyse. Tout le code nécessaire au fonctionnement de ces fenêtres reste à la charge du développeur. [11]

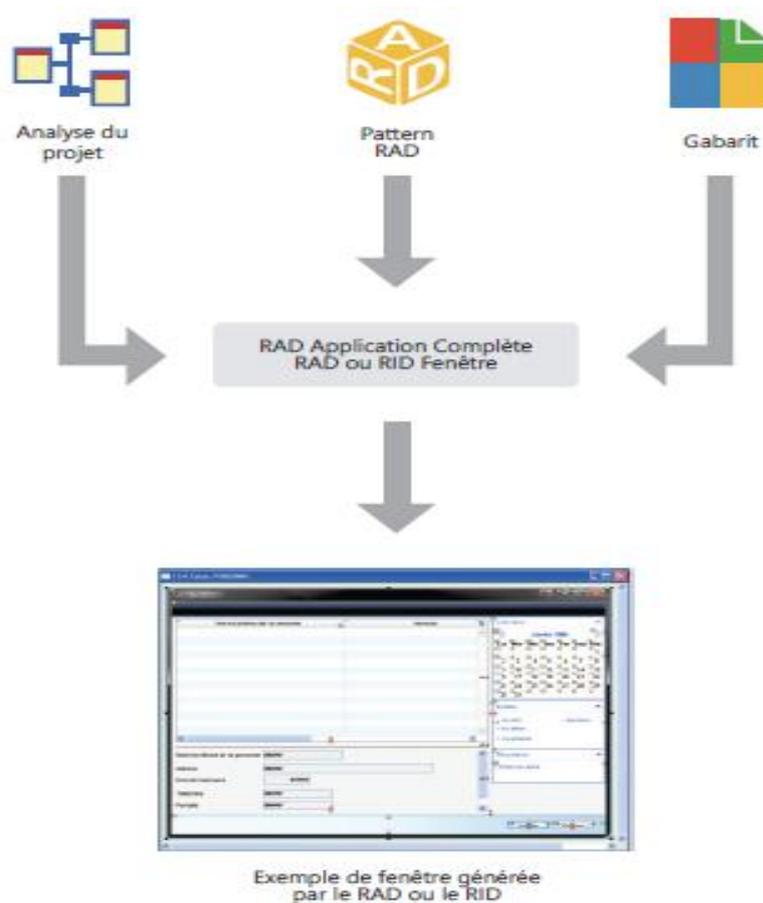


Figure 20 : Génération de fenêtre avec RAD ou RID

Conclusion

Dans ce chapitre nous avons recensé et présenté les éléments essentiels pour la réussite d'un projet de développement logiciel. Nous avons aussi présenté l'environnement de développement Windev Mobile et les différents outils qu'il offre pour développer une application mobile et gérer le projet de A à Z.

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

CHAPITRE 3 :

Etude de cas- Analyse des besoins, Prototypage et maquettage des IHM 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 l'analyse des besoins, maquettage et prototypage des IHM d'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 tâches comme suit :

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 1 : répartition des tâches

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

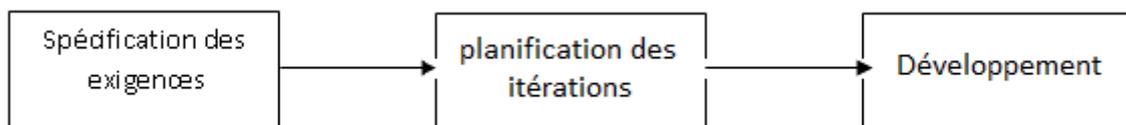


Figure 21 : Démarche adoptée

Nous avons planifié trois itérations :

- 1^{ère} itération : développer le noyau (version 1) qui contient les fonctionnalités essentielles.
- 2^{ème} itération : développer la version 2.
- 3^{ème} itération : développer la version finale qui contient toutes les fonctionnalités.

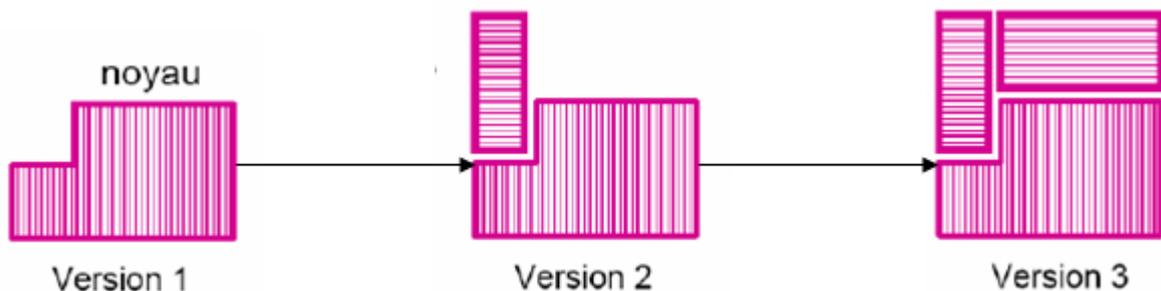


Figure 22 : Les différentes versions de notre application

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 développer l'application mobile 'ypiresia' pour la mise en relation du consommateur et du fournisseur du service.

L'application "ypiresia" doit permettre au fournisseur du service de créer un profil professionnel, de présenter ses services et localiser son activité.

Elle doit aussi faciliter au client de trouver le fournisseur de service, de consulter son profil, de le localiser et le contacter facilement et rapidement.

4. Acteurs de l'application

Nous avons identifié trois acteurs externes au système :

- **Le fournisseur de service** : qui crée un profil professionnel et propose son service dans le cadre de son activité.
- **Le client** : qui crée un compte utilisateur et cherche un service.
- **Le visiteur** : qui utilise l'application pour localiser un fournisseur de service.
-

5. Fonctionnalités de l'application

Exigence	Acteur
Créer un compte	Client, fournisseur
S'authentifier	Client, fournisseur
Rechercher un fournisseur par nom	Client, visiteur
Rechercher un fournisseur par service	Client, visiteur

Rechercher un fournisseur par adresse	Client, visiteur
Rechercher un fournisseur par carte	Client, visiteur
Rechercher un fournisseur à proximité	Client, visiteur
Localiser un fournisseur de service	Client, visiteur
Publier une demande d'un service	Client
publier une offre	Fournisseur
Afficher un itinéraire	Client, Visiteur
Enregistrer sa position	Fournisseur
Visualiser les informations d'un fournisseur	Client, Visiteur
Afficher les détails d'un fournisseur	Client
Contacteur un fournisseur	Client
Contacteur un client	Fournisseur
Gérer son compte	Client, Fournisseur
Se déconnecter	Client, Fournisseur

Tableau 2 : Tableau des exigences fonctionnelles

6. Planification itérative

On s'est basée sur une planification itérative pilotée par les exigences fonctionnelles et les risques.

Pour chaque itération on a choisi les cas d'utilisations présentant ces trois qualités :

- Significatif du point de vue de l'architecture.
- De priorité fonctionnelle importante.
- A haut risque.

La condition d'arrêt de chaque itération est la satisfaction de client.

On a donc hiérarchisé les exigences selon trois niveaux qui définissent les 3 incréments à développer selon le tableau suivant :

Exigence	Priorité	Risque	Itération
Créer un compte	Haute	Moyen	1#
S'authentifier	Haute	Haut	1#
Rechercher un fournisseur par nom	Haute	Faible	1#
Rechercher un fournisseur par service	Haute	Faible	1#
Rechercher un fournisseur par adresse	Haute	Faible	1#
Rechercher un fournisseur par carte	Moyenne	Faible	2#
Rechercher un fournisseur à proximité	Faible	Faible	3#
Localiser un fournisseur de service	Moyenne	Faible	2#
Publier une demande d'un service	Moyenne	Faible	2#
publier une offre	Faible	Faible	3#
Afficher un itinéraire	Moyenne	Faible	2#
Enregistrer sa position	Moyenne	Moyen	2#
Visualiser les informations d'un fournisseur	Haute	Faible	1#
Afficher les détails d'un fournisseur	Haute	Faible	1#
Contacteur un fournisseur	Moyenne	Faible	2#
Contacteur un client	Faible	Faible	3#
Gérer son compte	Haute	Moyen	1#
Se déconnecter	Haute	Faible	1#

Tableau 3 : Tableau des itérations

7. Analyse des besoins

7.1 Cycle de vie d'une itération

Pour affiner les besoins et avoir une vision détaillée des cas d'utilisations nous avons opté pour le cycle illustré par la figure 23.

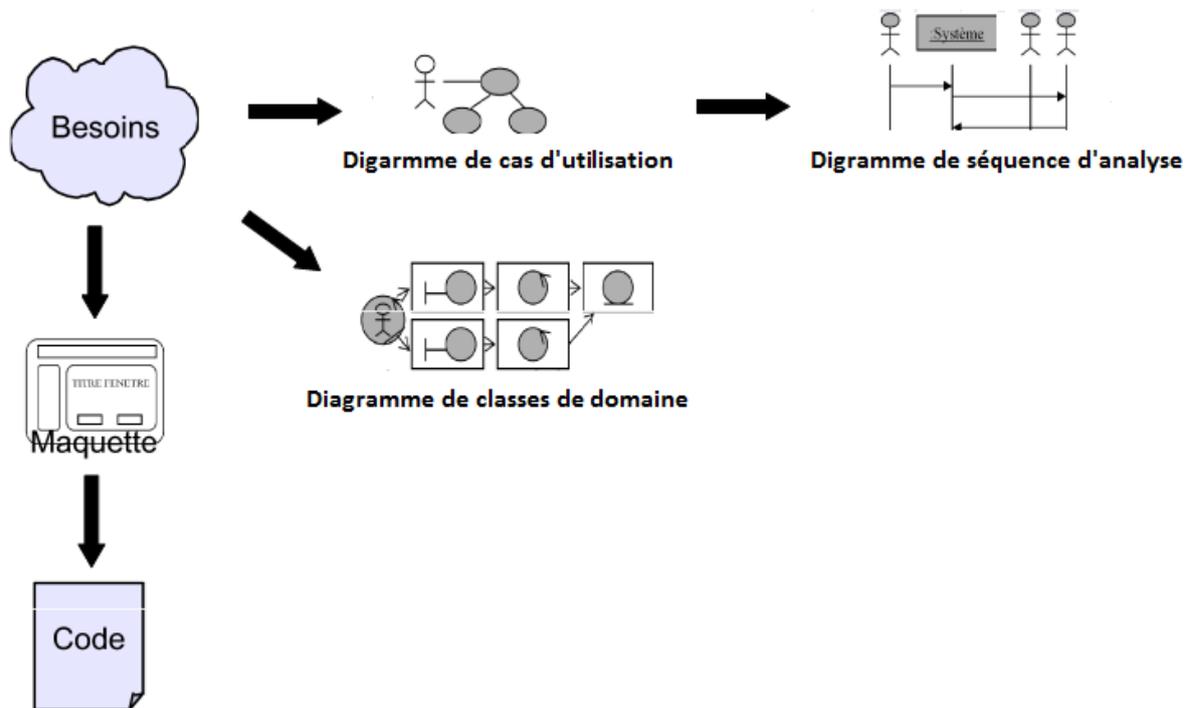


Figure 23: Cycle de vie d'une itération

A partir des besoins qu'on a déterminé nous réaliserons les diagrammes de cas d'utilisation de notre application mobile "ypiresia" et les diagrammes de séquence boîte noire correspond à chaque cas d'utilisation. Ensuite nous présentons les différentes maquettes de l'application, et à la fin nous donnons le diagramme de classe d'analyse correspond au système de l'application.

7.2 Diagrammes de cas d'utilisation

Nous avons utilisé les diagrammes de cas d'utilisation pour donner une vision globale du comportement fonctionnel de notre application et pour décrire l'interaction entre les acteurs et le système. Nous avons réalisés trois diagrammes, chaque diagramme est associé à son acteur (Client, Fournisseur de

service et Visiteur), et à la fin nous les avons rassemblés dans un diagramme global

7.2.1 Diagramme de cas d'utilisation associé au client

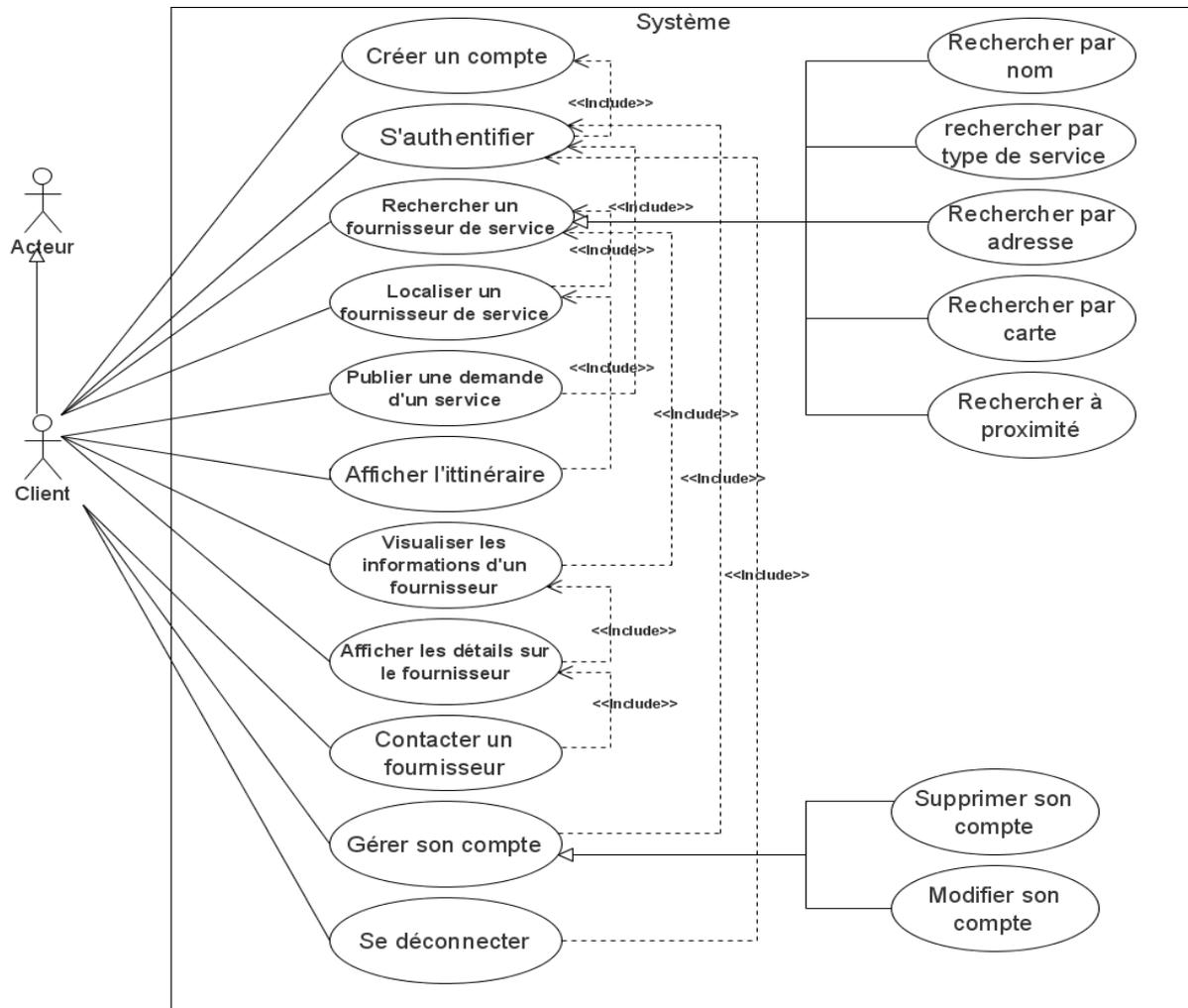


Figure 24 : Diagramme de cas d'utilisation associé au client

7.2.2 Diagramme de cas d'utilisation associé au fournisseur

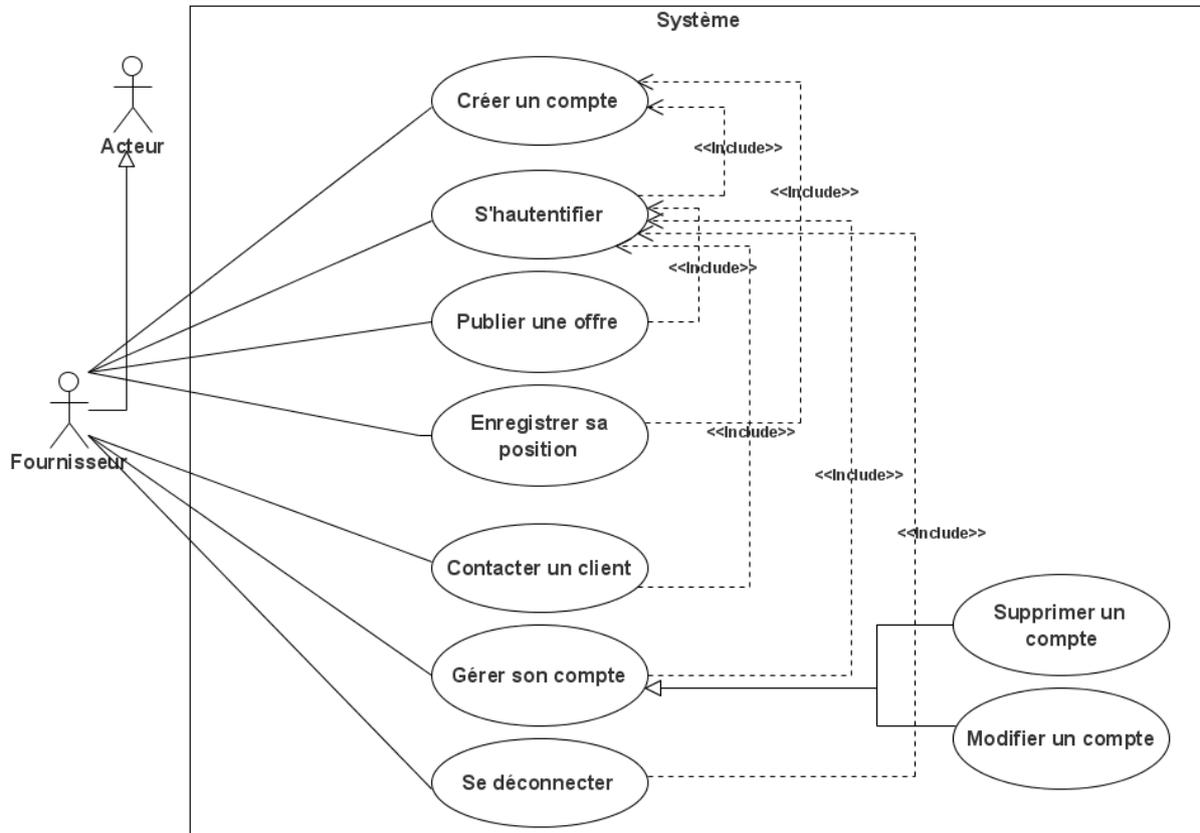


Figure 25 : Diagramme de cas d'utilisation associé au fournisseur

7.2.3 Diagramme de cas d'utilisation associé au visiteur

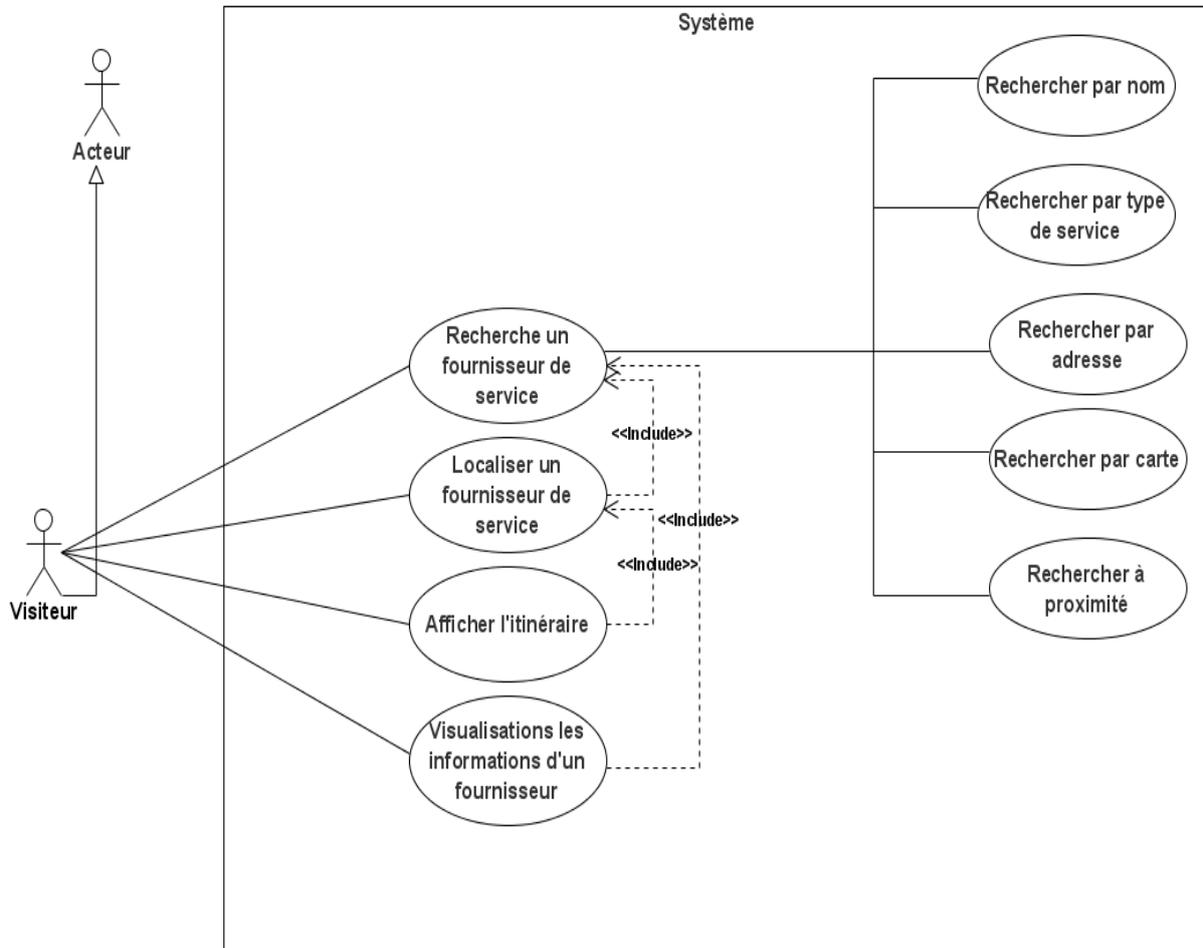


Figure 26 : Diagramme de cas d'utilisation associé au visiteur

7.2 .4 Diagramme de cas d'utilisation global

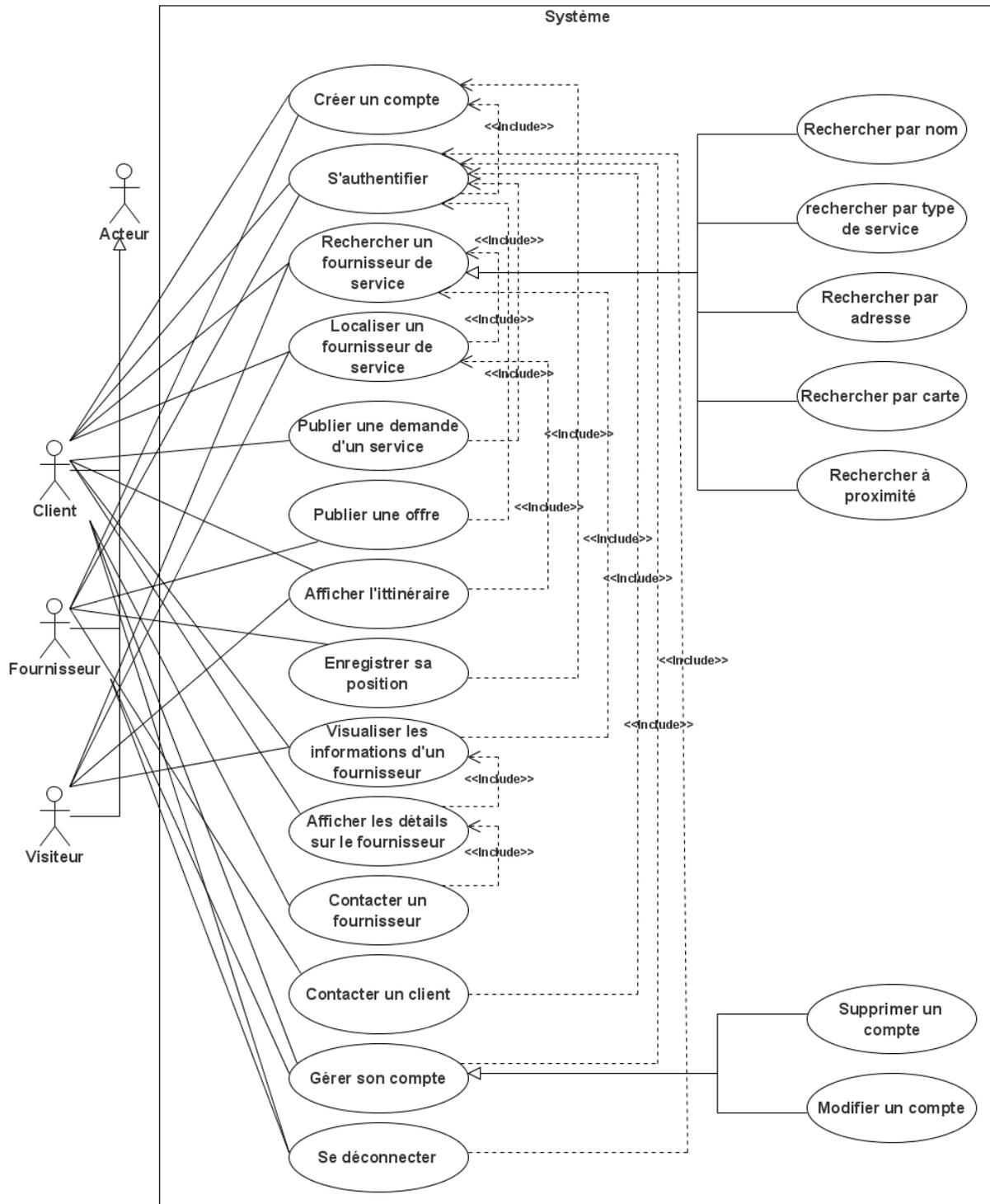


Figure 27 : Diagramme de cas d'utilisation global

7.3 Prototypage et maquettage de l'application

7.3.1 Le maquettage

Le maquettage permet en particulier de lever les incertitudes liées à la conception des SI, en facilitant la démarche de planification qui précède la conception et la réalisation.

Il s'agira d'une suite d'écrans montrant la navigation à travers l'application ainsi que son utilisation. Cette maquette doit pouvoir refléter les besoins des clients et donc doit être testée et avalisée par ces derniers. Et enfin, elle doit pouvoir mettre en avant les difficultés majeures pour les développeurs, ce qui leurs permettra de recentrer leur effort

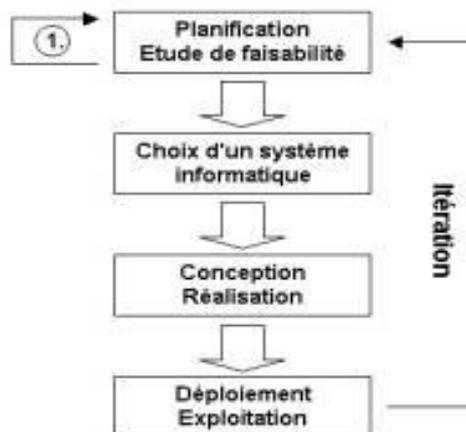


Figure 28 :Réévaluation par maquettage.

7.3.2 Le prototypage

Le prototypage est la création de maquettes simples et incomplètes d'un design pour explorer des idées, élaborer le besoin, affiner les spécifications et tester les fonctionnalités. Il y a trois types basiques de prototypage : concept, jetable et avec évolution.

Le prototypage permet d'aborder par étapes successives la conception et la réalisation d'applications, en utilisant les « feedbacks » des utilisateurs. Il s'agira de versions informatiques testées par les développeurs et les utilisateurs finaux pour vérifier que l'application répond à leur attente.

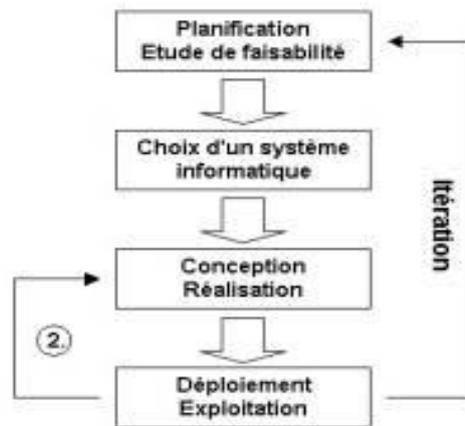


Figure 29 :Réévaluation par prototypage.

7.3.3. A quel moment maquette-t-on ?

- Le prototypage peut être employé tout au long du cycle de développement. Il est recommandé d'y recourir le plus en amont possible, avant le début de tout développement informatique !
- Des versions successives de prototypes peuvent être prévues, s'intégrant dans une démarche de conception itérative.
- Le degré de précision du prototype est déterminé par le stade du cycle de développement du projet.
- Cependant, on ne peut commencer les phases de conception qu'une fois l'architecture du site réfléchi :
 - choix des catégories.
 - choix des pages.
 - réflexion sur la navigation.
 - type de contenu.
 - objectif de site.

7.3.4 Classification des prototypes

Il existe différents types de prototypes, qui peuvent être classés selon 4 critères :

- La pérennité
- L'interactivité
- Le degré de fidélité
- Le support

A. La pérennité

- **Le prototypage « jetable »** est utilisé pour tester certains aspects de l'application, sans but de récupération pour l'application finale.
- **Le prototypage « itératif »** développe plusieurs versions successives d'une interface qui serviront de base à l'application finale.
- **Le prototypage « incrémental »** consiste à développer et tester progressivement différentes parties de l'application, jusqu'à sa réalisation complète.

B. L'interactivité

- **Le prototypage « statique » (ou horizontal)** : présente une vision globale de l'interface, mais sans permettre d'interaction directe avec l'utilisateur. Il permet aussi de vérifier que les utilisateurs comprennent bien la structure et l'organisation du système.
- **Le prototypage « interactif » (ou vertical)** : couvre seulement certains aspects mais de façon complète et opérationnelle. Ex : processus de commande en ligne. Il permet aussi d'effectuer des tests d'utilisabilité.

L'interactivité peut être simulée sur support papier.

C. Le degré de fidélité

- **Le prototype « basse fidélité »** : il s'agit le plus souvent d'une maquette papier qui facilite l'implication et la concentration de l'utilisateur sur les aspects fonctionnels (évite les dérives sur l'esthétique).

Il peut être annoté en cours de test, est coûteux, et permet de tester tôt le concept de l'interface.

Le prototype « haute fidélité » : il s'agit d'un prototype interactif, sur ordinateur, qui présente un aspect graphique très proche de l'interface finale.

- Offre à l'utilisateur une vision réaliste de l'interface même si tout n'est pas intégralement implémenté.
- Intervient généralement à un stade avancé du développement.

D. Le support

- **Le prototype papier** : le prototype est présenté sur papier, les différents écrans sont esquissés présentant le positionnement des zones fonctionnelles principales.

Le rendu manuel du prototype a un impact pour l'utilisateur : plus le trait est technique, plus l'utilisateur tend à considérer que le concept est finalisé et à s'autocensurer dans ses commentaires.

- **Le prototype électronique** : le prototype est présenté sur écran, sous forme statique ou interactive et le choix de l'outil dépend de l'interactivité souhaitée.
- **Le prototype vidéo** : très proche du prototype papier dans sa fonctionnalité, ce prototype est réalisé en filmant le fonctionnement de l'interface (par maquette papier ou capture d'écran), ici pas d'interaction directe mais une meilleure représentation du fonctionnement de l'interface.

7.3.5 Importance du maquettage/prototypage dans le processus de conception d'IHM

Le maquettage / prototypage est un élément essentiel d'une conception itérative d'IHM et centrée sur la prise en compte des usages.

En ergonomie, le maquettage/ prototypage consiste à simuler l'interface d'un produit, de façon complète ou partielle, afin d'obtenir des informations sur l'interaction des utilisateurs avec le futur produit. Effectué suffisamment tôt, le prototypage permet de détecter les problèmes d'utilisabilité et de vérifier les besoins des utilisateurs avant le codage complet. Cela favorise l'optimisation du processus et des coûts de développement.

Une fois qu'on a compris les utilisateurs cibles, leurs contraintes et leurs besoins, on va pouvoir se projeter et imaginer ce que pourrait être le produit. Le maquettage donne la possibilité d'avoir un feedback au sein du projet ou d'utilisateurs potentiels (tests utilisateurs sur maquettes). Le maquettage permet d'optimiser les différents types de maquette jusqu'à ce que la performance du produit ou sa qualité ergonomique soit atteinte.

7.3.6 Les avantages de maquettages

Les avantages du maquettage sont :

- L'usage de ces outils de maquettage ou de prototypage est primordial et sert de base à la phase de conception d'un projet d'une application. Les avantages sont multiples :
- cela favorise la définition du périmètre fonctionnel.
- cela permet de présenter aux utilisateurs des éléments sur lesquels ils vont pouvoir réagir.
 - cela permet de se concentrer sur le fond (fonctionnalités, interactions, contenus), sans se laisser distraire par la forme (design).
 - cela facilite l'évolution de l'interface proposée au départ, afin qu'elle corresponde le mieux possible aux attentes et besoins des futurs utilisateurs.

- cela permet, au besoin, de corriger ou valider des choix (techniques, fonctionnels) avant de commencer le développement technique.
- l'intervention d'un ergonomiste permet d'obtenir des retours d'expérience et des bonnes pratiques ergonomiques.
- Incitent à la critique et la favorisent.
- Peu coûteux et peuvent être de très bas niveau de fidélité.
- Répondent au besoin d'obtenir des résultats rapides à des tests lors de la conception.
- Orientent la discussion lors d'un atelier.
- Permettent à tous les acteurs du projet de partager une même représentation du produit vers lequel tendre.

Dans notre application mobile nous avons utilisé le principe de prototypage itératif qui constitue un élément-clé du développement itératif, il consiste à Concevoir des versions intermédiaires et incomplètes d'un logiciel, conçues pour tester l'utilisation avant la phase de programmation.

Nous avons utilisé le prototypage itératif pour appréhender rapidement et à moindre coût les attentes et les réactions des utilisateurs. Nous nous sommes basés sur la création de prototypes successifs et améliorés en fur à mesure du développement comme illustré sur la figure 30.[12]

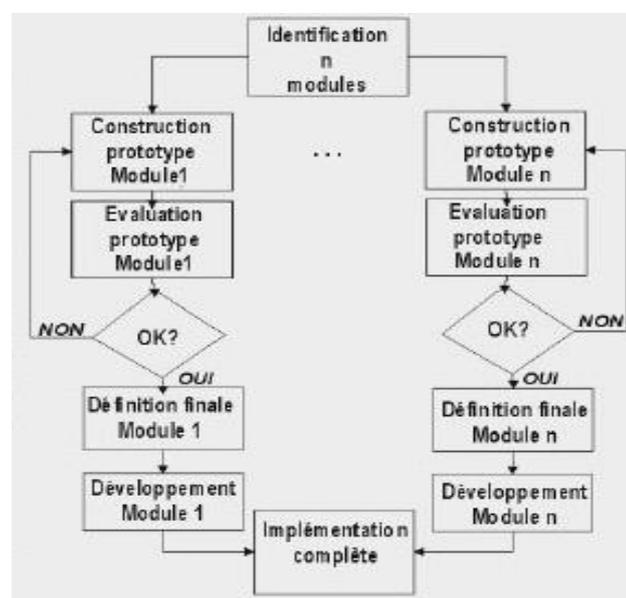


Figure 30 : cycle de prototypage de l'application ypiresia

Nous avons construit les prototypes de l'application 'ypiresia' sur Windev mobile en utilisant les ancrages et les modèles de fenêtres.

Les figures suivantes représentent quelques maquettes réalisées, pour chaque maquette nous allons décrire leurs caractéristiques et leurs rôles dans notre application ypiresia :

7.4. Maquettes de l'application Ypiresia

7.4.1 Maquette d'authentification

Cette maquette correspond à l'authentification par la saisie du pseudo et du mot de passe.

Le lien « Je n'ai pas de compte » permet d'inscrire et avoir un compte.

Le lien « J'accède comme visiteur » permet d'accéder à l'application juste comme un visiteur. Le bouton « Se connecter » permet de valider l'authentification.



Authentification

logo
d'application

Pseudo

mot de passe

Se connecter

J'accède comme visiteur

je n'ai pas de compte

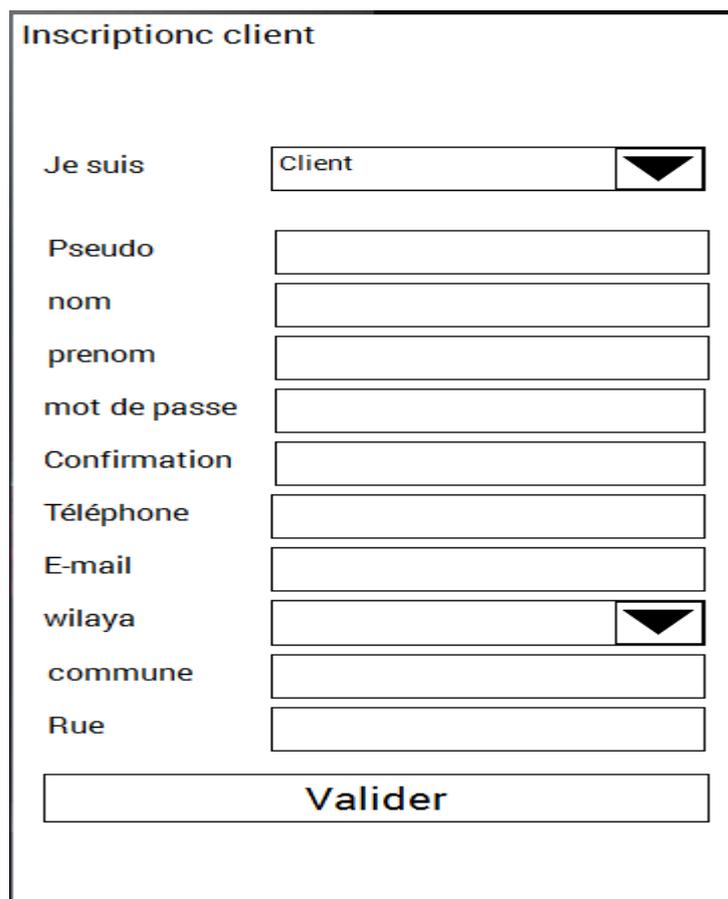
Figure 31 : Maquette de la fenêtre d'authentification

7.4.2 Maquette d'inscription d'un client

Cette maquette correspond à l'inscription d'un client dans l'application par remplissage de l'ensemble des champs.

Le bouton « Valider » permet de valider l'inscription d'un client.

Le combo box à côté du libellé « Je suis » permet de choisir l'inscription d'un client.



Inscriptionc client

Je suis	<input type="text" value="Client"/>	▼
Pseudo	<input type="text"/>	
nom	<input type="text"/>	
prenom	<input type="text"/>	
mot de passe	<input type="text"/>	
Confirmation	<input type="text"/>	
Téléphone	<input type="text"/>	
E-mail	<input type="text"/>	
wilaya	<input type="text"/>	▼
commune	<input type="text"/>	
Rue	<input type="text"/>	
<input type="button" value="Valider"/>		

Figure 32 : Maquette de la Fenêtre inscription client

7.4.3 Maquette d'inscription personne morale

Cette maquette correspond à l'inscription d'une personne morale dans l'application par remplissage de l'ensemble des champs.

Le bouton « Valider » permet de valider l'inscription d'une personne morale.

Le combo box à côté du libellé « Je suis » permet de choisir l'inscription d'une personne morale.

Inscription personne morale

Je suis ▼

pseudo

Réseau social

mot passe

confirmation

telephone

E-mail

Activité ▼

wilaya ▼

commune

Rue

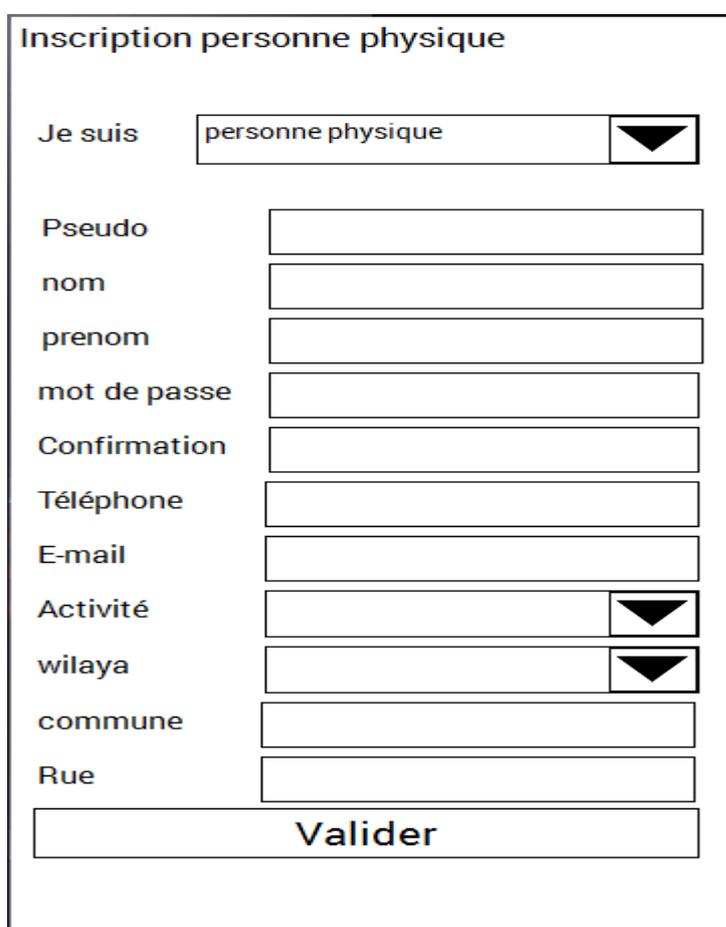
Figure 33 : Maquette de la Fenêtre inscription personne morale

7.4.4 Maquette d'inscription personne physique

Cette maquette correspond à l'inscription d'une personne physique par remplissage de l'ensemble des champs.

Le bouton « Valider » permet de valider l'inscription d'une personne physique.

Le combo box à côté du libellé « Je suis » permet de choisir l'inscription d'une personne physique.



Maquette de la Fenêtre inscription personne physique. Le formulaire est intitulé "Inscription personne physique" et contient les champs suivants :

- Je suis : (combo box)
- Pseudo :
- nom :
- prenom :
- mot de passe :
- Confirmation :
- Téléphone :
- E-mail :
- Activité : (combo box)
- wilaya : (combo box)
- commune :
- Rue :

Le bouton "Valider" est situé en bas du formulaire.

Figure 34 : Maquette de la Fenêtre inscription personne physique

7.4.5 Maquette correspondante à la rubrique paramètres du profil client

Cette maquette correspond à la rubrique « paramètres » du profil client.

Le bouton « Modifier mon compte » permet au client de modifier son compte.
Le bouton « Supprimer mon compte » permet au client de supprimer son compte.

Le bouton « Ajouter des détails » permet au client d'ajouter des détails à son compte.

Le bouton « Publier une demande » permet au client de publier une demande de service.

Le bouton « Déconnexion » permet au client de se déconnecter.

Paramètres	Offres	Recherche	Carte
photo de profil	pseudo		
Modifier mon compte			
Supprimer mon compte			
Ajouter des détails			
Publier une demande			
Déconnexion			

Figure 35 : Maquette correspondante à la rubrique paramètres du compte client

7.4.6 Maquette correspondante à la rubrique Offres du profil client

Cette maquette correspond à la rubrique « Offres » qui affiche les offres publiées par les fournisseurs de service. Chaque offre contient sa description, sa date et l'heure de publication. L'application n'affiche que les offres récentes.

Paramètres	Offres	Recherche	Carte
<p>pseudo</p> <div style="border: 1px solid black; padding: 5px; min-height: 40px;">description</div> <p style="text-align: right;">date heure</p> <hr/>			
<p>pseudo</p> <div style="border: 1px solid black; padding: 5px; min-height: 40px;">description</div> <p style="text-align: right;">date heure</p> <hr/>			
<p>pseudo</p> <div style="border: 1px solid black; padding: 5px; min-height: 40px;">description</div> <p style="text-align: right;">date heure</p>			

Figure 36 : Maquette correspondante à la rubrique offres du profil client

7.4.7. Maquette correspondante à la rubrique Recherche du profil client

Cette maquette correspond à la rubrique « Recherche » du profil client qui peut réaliser sa recherche selon le nom de fournisseur, l'activité et l'adresse (wilaya, commune).

Le Combo Box permet de choisir la wilaya et le type d'activité correspondante à la recherche.

Le bouton « Filtrer » permet de filtrer les résultats de la recherche selon le nom, la wilaya, commune et activité.

Paramètres	Offres	Recherche	Carte
Nom		<input type="button" value="botton de recherche"/>	
Sélectionnez votre type de recherche			
Activité		<input type="button" value="▼"/>	
Wilaya		<input type="button" value="▼"/>	
commune			
<input type="button" value="Filtrer"/>			

Figure 37 : Maquette correspondante à la rubrique recherche du profil client

7.4.8. Maquette correspondante à la rubrique Recherche du profil Fournisseur

Cette maquette correspond à la rubrique « Paramètres » du profil fournisseur,.

Paramètres	Demandes	Carte
	pseudo	
Modifier mon compte		

Supprimer mon compte		

Ajouter des détails		

Publier une offre		

Déconnexion		

Figure 38 : Maquette correspondante à rubrique paramètres du profil fournisseur

7.4.9 Maquette correspondante à la rubrique Demandes du profil Fournisseur

Cette maquette correspond à la rubrique « Demandes » du profil Fournisseur. Cette rubrique permet d'afficher les demandes de services publiées par les clients. Chaque demande contient sa description, sa date et l'heure de la publication. L'application n'affiche au Clients que les demandes récentes.

Paramètres	Demandes	Carte
pseudo		
<div style="border: 1px solid black; padding: 5px;">description</div>		
date heure		
<hr/>		
pseudo		
<div style="border: 1px solid black; padding: 5px;">description</div>		
date heure		
<hr/>		
pseudo		
<div style="border: 1px solid black; padding: 5px;">description</div>		
date heure		

Figure 39 : Maquette correspondante à la rubrique demandes du profil fournisseur

7.4.10. Maquette de la fenêtre résultats de la recherche

Cette maquette représente les résultats de la recherche réalisée par un client. Le fournisseur de service qui peut-être une personne ou une société. Cette maquette nous montre une liste de tous les fournisseurs dans les volets « Personne », « Société » et leurs détails associés.

Résultats de la recherche	
Personne	Société
Photo	pseudo Wilaya Sevice
Photo	pseudo Wilaya Sevice
Photo	pseudo Wilaya Sevice

Figure 40 : Maquette de la fenêtre résultats de la recherche.

7.4.11 Maquette de la fenêtre détails de la recherche

Cette maquette correspond aux détails des résultats de la recherche réalisée par le client, qui donne toutes les informations sur le fournisseur recherché.

détails de la recherche	
pseudo_p	<input type="text"/>
nom	<input type="text"/>
wilaya	<input type="text"/>
commune	<input type="text"/>
rue	<input type="text"/>
tel	<input type="text"/>
email	<input type="text"/>
service	<input type="text"/>
Compte_facebook	<input type="text"/>

Figure 41 : Maquette de la fenêtre détails de la recherche

7.4.12 Maquette de la fenêtre modifier compte client

Cette maquette correspond à la modification des informations du compte client.

Le bouton « Annuler » permet d'annuler les modifications et garder les anciennes informations.

La maquette de la fenêtre 'modifier compte client' est présentée dans un cadre rectangulaire. Elle est structurée de la manière suivante :

- En haut à gauche, le titre **modifier compte client** est écrit.
- Juste en dessous, un rectangle blanc est centré et contient le texte **photo de profil**.
- En dessous de ce rectangle, le mot **Pseudo** est centré.
- Une ligne horizontale sépare le pseudo des autres sections.
- Le mot **informations personnelles** est centré.
- Une autre ligne horizontale sépare les informations personnelles de la confidentialité.
- Le mot **confidentialité** est centré.
- Une troisième ligne horizontale sépare la confidentialité de l'adresse.
- Le mot **adresse** est centré.
- Une quatrième ligne horizontale sépare l'adresse du contact.
- Le mot **contact** est centré.
- Une cinquième ligne horizontale sépare le contact de la localisation.
- Le mot **Localisation** est centré.
- Une dernière ligne horizontale sépare la localisation du bouton d'annulation.
- En bas, un grand rectangle blanc est centré et contient le mot **Annuler**.

Figure 42 : Maquette de la Fenêtre modifier compte client

7.4.13 Maquette de la fenêtre modifier le compte personne morale

Cette maquette correspond à la modification des informations du compte d'une personne morale.

Le bouton « Annuler » permet d'annuler les modifications et garder les anciennes informations.

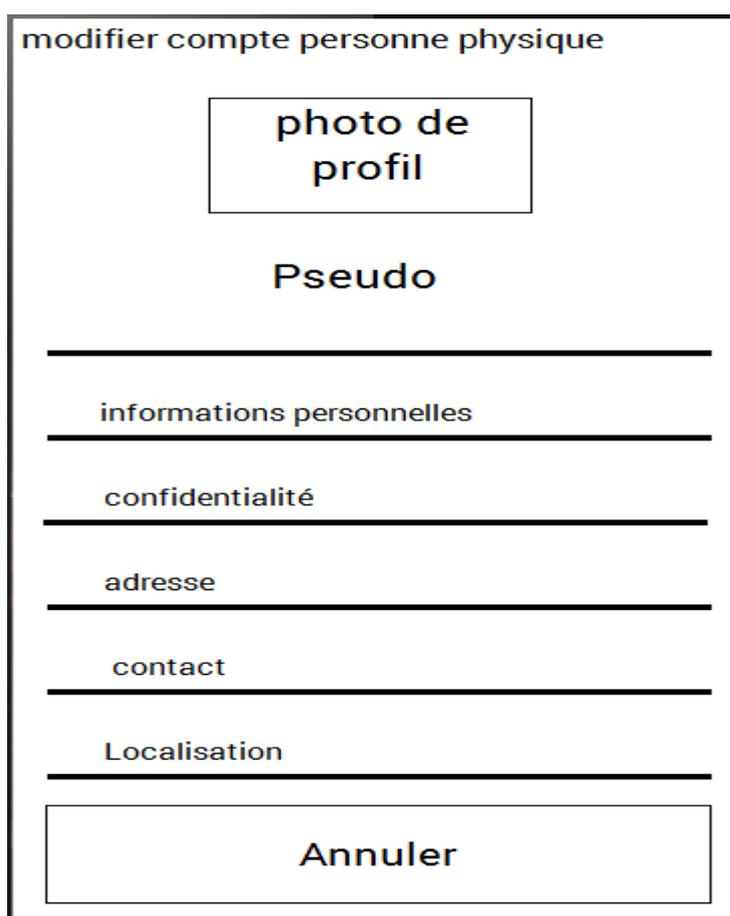
La maquette est présentée dans un cadre rectangulaire. À l'intérieur, le titre 'modifier compte personne morale' est écrit en haut à gauche. Au centre, il y a un rectangle blanc avec le texte 'photo de profil'. En dessous, le mot 'Pseudo' est centré. Une ligne horizontale sépare 'Pseudo' des sections suivantes. Ces sections sont : 'informations personnelles', 'confidentialité', 'adresse', 'contact' et 'Localisation', chacune précédée d'une ligne horizontale. À la base de la maquette, un grand rectangle blanc contient le bouton 'Annuler'.

Figure 43 : Maquette de la Fenêtre modifier compte personne moral

7.4.14. Maquette de la fenêtre modifier le compte d'une personne physique

Cette maquette correspond à la modification des informations du compte personne physique.

Le bouton « Annuler » permet d'annuler les modifications et garder les anciennes informations.



modifier compte personne physique

photo de profil

Pseudo

informations personnelles

confidentialité

adresse

contact

Localisation

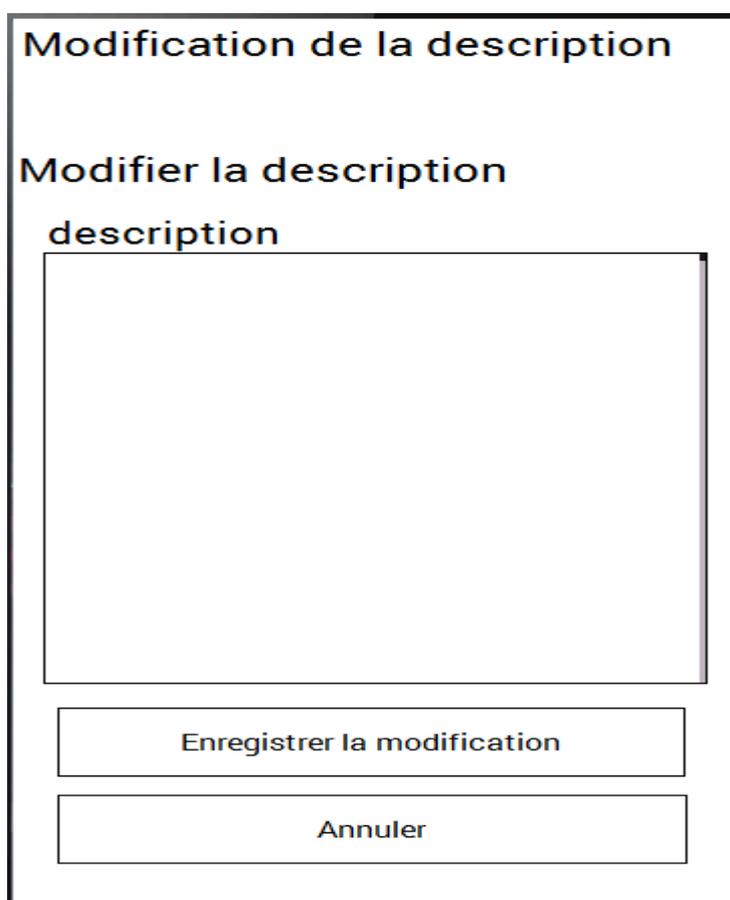
Annuler

Figure 44 : Maquette de la Fenêtre modifier compte personne physique

7.4.15. Maquette de la fenêtre modifier une description

Cette maquette correspond à la modification de la description d'une offre d'un fournisseur de service ou bien une demande de service d'un client.

On trouve deux boutons « Enregistrer la modification » pour enregistrer les modifications réalisées et le bouton « Annuler » pour annuler les actions précédentes.



Maquette de la fenêtre "Modification de la description".

Le titre de la fenêtre est "Modification de la description".

Le contenu principal de la fenêtre est "Modifier la description", suivi d'un champ de saisie "description".

En dessous du champ de saisie, il y a deux boutons : "Enregistrer la modification" et "Annuler".

Figure 45 : Maquette de la Fenêtre modifier la description

7.4.16 Maquette de la fenêtre modifier adresse

Cette maquette correspond à la modification de l'adresse d'un client ou d'un fournisseur de service.

Le combo box « wilaya » permet de choisir la wilaya.

Le champ « Commune » correspond aux communes de la wilaya sélectionnée et le champ « rue » à la rue.

On trouve deux boutons « Enregistrer la modification » pour enregistrer les modifications réalisées et « Annuler » pour annuler la modification.

Maquette de la fenêtre 'Modification d'adresse' :

- Titre : **Modification d'adresse**
- Section : **Modifier mon adresse**
- Champs de saisie :
 - Wilaya : Champ de saisie avec une flèche descendante à droite.
 - Commune : Champ de saisie.
 - Rue : Champ de saisie.
- Boutons :
 - Enregistrer la modification
 - Annuler

Figure 46 : Maquette de la fenêtre modifier adresse

7.4.17. Maquette de la fenêtre modifier un compte Facebook

Cette maquette correspond à la modification du compte Facebook d'un client ou d'un fournisseur de service.

Modification de compte facebook

Modifier le compte facebook

Figure 47 : Maquette de la fenêtre modifier le compte Facebook

7.4.18. Maquette de la fenêtre modifier le mot de passe

Cette maquette correspond à la modification du mot de passe d'un compte client ou d'un compte fournisseur de service. Il suffit de saisir l'ancien mot de passe, entrer le nouveau et le confirmer. A la fin un client ou un fournisseur doit enregistrer le nouveau mot de passe ou bien de l'annuler et de garder l'ancien.

Modification de mot passe

Modifier le mot de passe

Ancien mot de passe

Nouveau mot de passe

Confirmation

Figure 48 : Maquette de la fenêtre modifier le mot de passe

7.4.19. Maquette de la fenêtre modifier le numéro de téléphone

Cette maquette correspond à la modification du numéro de téléphone d'un client ou d'un fournisseur de service, par la saisie du nouveau numéro de téléphone.

Maquette de la fenêtre 'Modification de numero du téléphone'. Le titre de la fenêtre est 'Modification de numero du téléphone'. Le contenu principal est 'Modifier le numéro de téléphone'. Il y a un champ de saisie pour le 'Numéro du téléphone'. En dessous, il y a deux boutons : 'Enregistrer la modification' et 'Annuler'.

Figure 49 : Maquette de la fenêtre modifier le numéro de téléphone

7.4.20 Maquette de la fenêtre modifier l'E-mail

Modification de l'e-mail

Modifier l'e-mail

Figure 50 : Maquette de la fenêtre modifier mon l'e-mail

7.4.21. Maquette de la fenêtre modifier le pseudo

Modification du pseudo

Modifier le pseudo

Figure 51 : Maquette de la fenêtre modifier le pseudo

7.4.22 Maquette de la fenêtre modifier nom

Modification de nom

Modifier le nom

Figure 52 : Maquette de la fenêtre modifier le nom

7.4.23 Maquette de la fenêtre modifier le prénom

Maquette de la fenêtre 'Modification de prénom'. Le titre de la fenêtre est 'Modification de prénom'. À l'intérieur, il y a un sous-titre 'Modifier le prénom', un champ de saisie contenant le mot 'Prenom', un bouton 'Enregistrer la modification' et un bouton 'Annuler'.

Figure 53 : Maquette de la fenêtre modifier le prénom

8. Diagrammes de séquences boîte noire

Nous avons décrit le comportement du système ‘ypiresia’ par des diagrammes de séquence système (DSS) où le système est vu comme une « boîte noire ». Le système est donc vu de l'extérieur (par les acteurs) sans préjuger de comment il sera réalisé. La « boîte noire » sera ouverte (décrite) seulement en conception

8.1. Diagramme de séquences du cas d'utilisation "créer un compte"

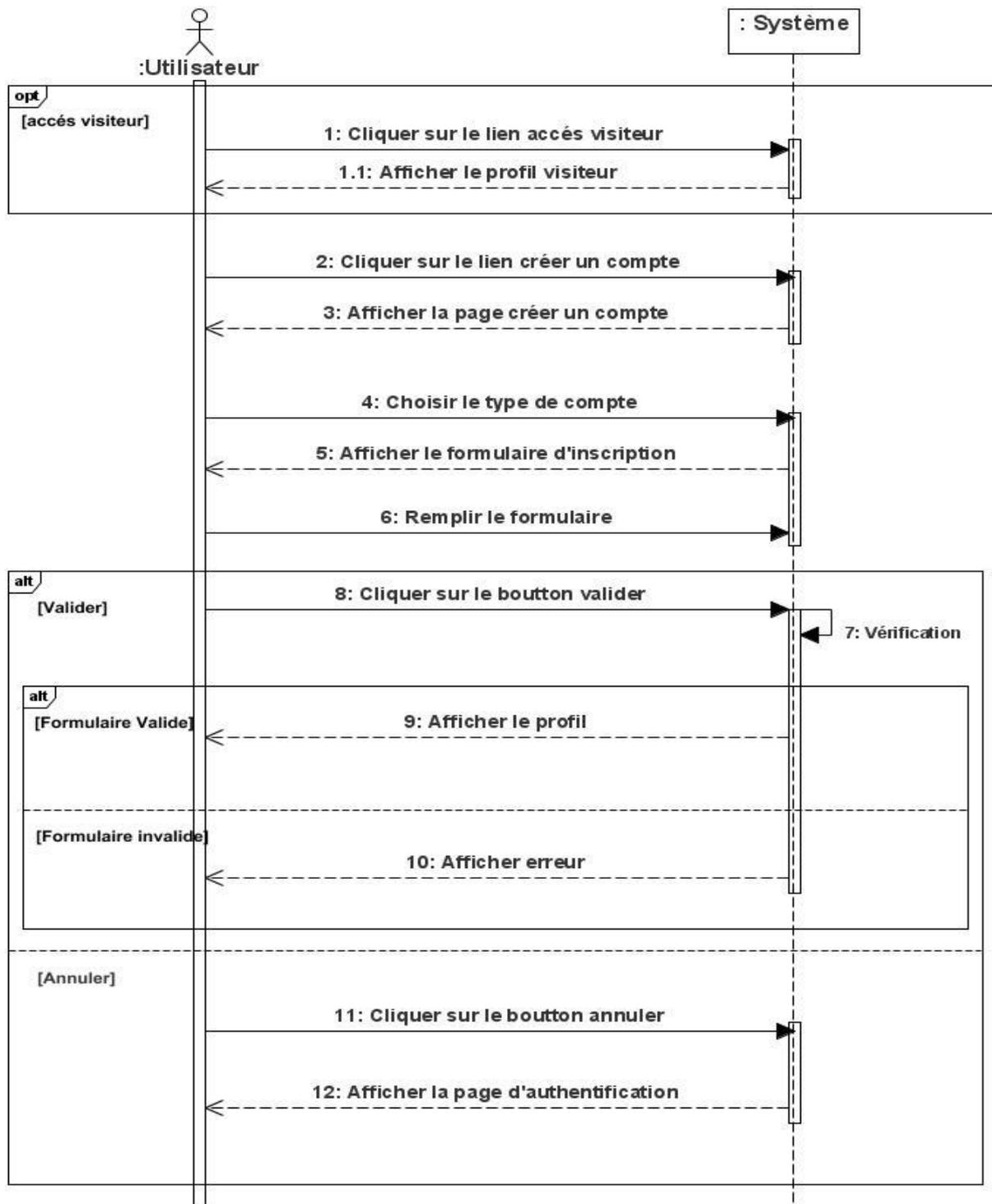


Figure 54 : Diagramme de séquences boîte noire du cas d'utilisation "créer un compte"

8.2. Diagramme de séquences du cas d'utilisation “s’authentifier”

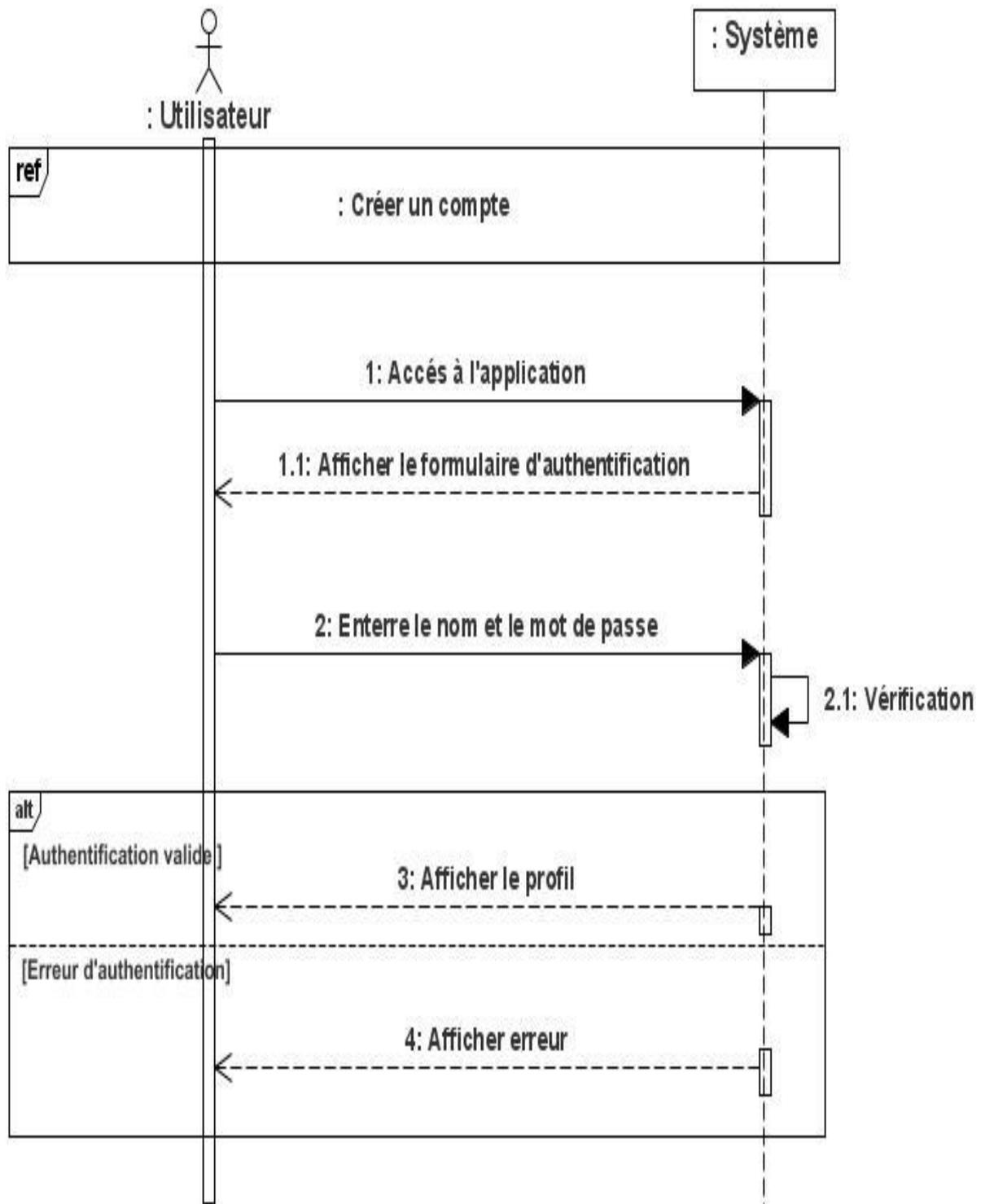


Figure 55 : Diagramme de séquences boîte noire du cas d'utilisation “s’authentifier”

8.3. Diagramme de séquence du cas d'utilisation “rechercher un fournisseur par nom”

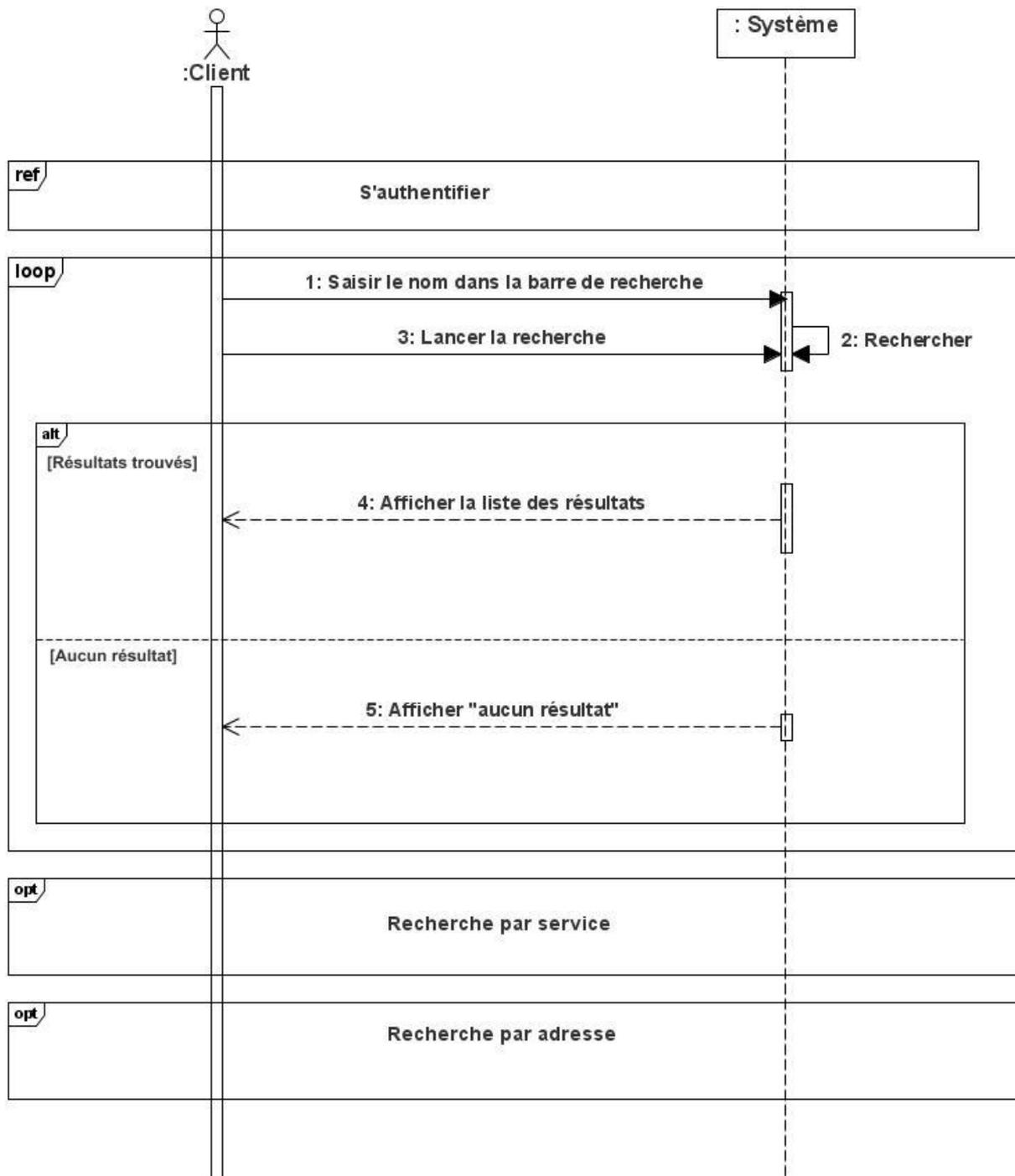


Figure 56 : Diagramme de séquences boîte noire du cas d'utilisation “rechercher un fournisseur par nom”

8.4. Diagramme de séquences du cas d'utilisation ‘rechercher un fournisseur par adresse’

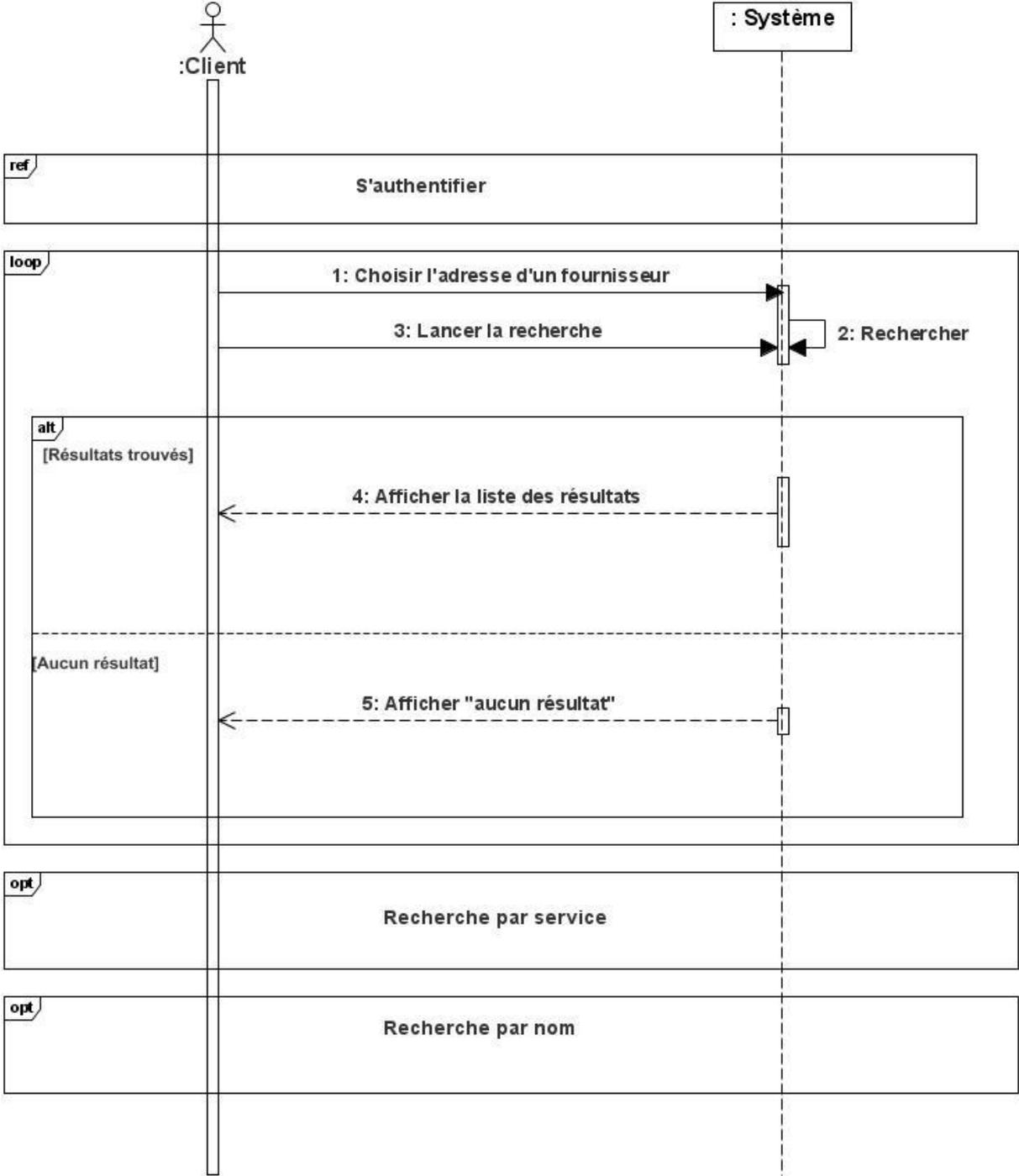


Figure 57 : Diagramme de séquences boîte noire du cas d'utilisation ‘rechercher un fournisseur par adresse’

8.5 Diagramme de séquences du cas d'utilisation ‘rechercher un fournisseur par service’

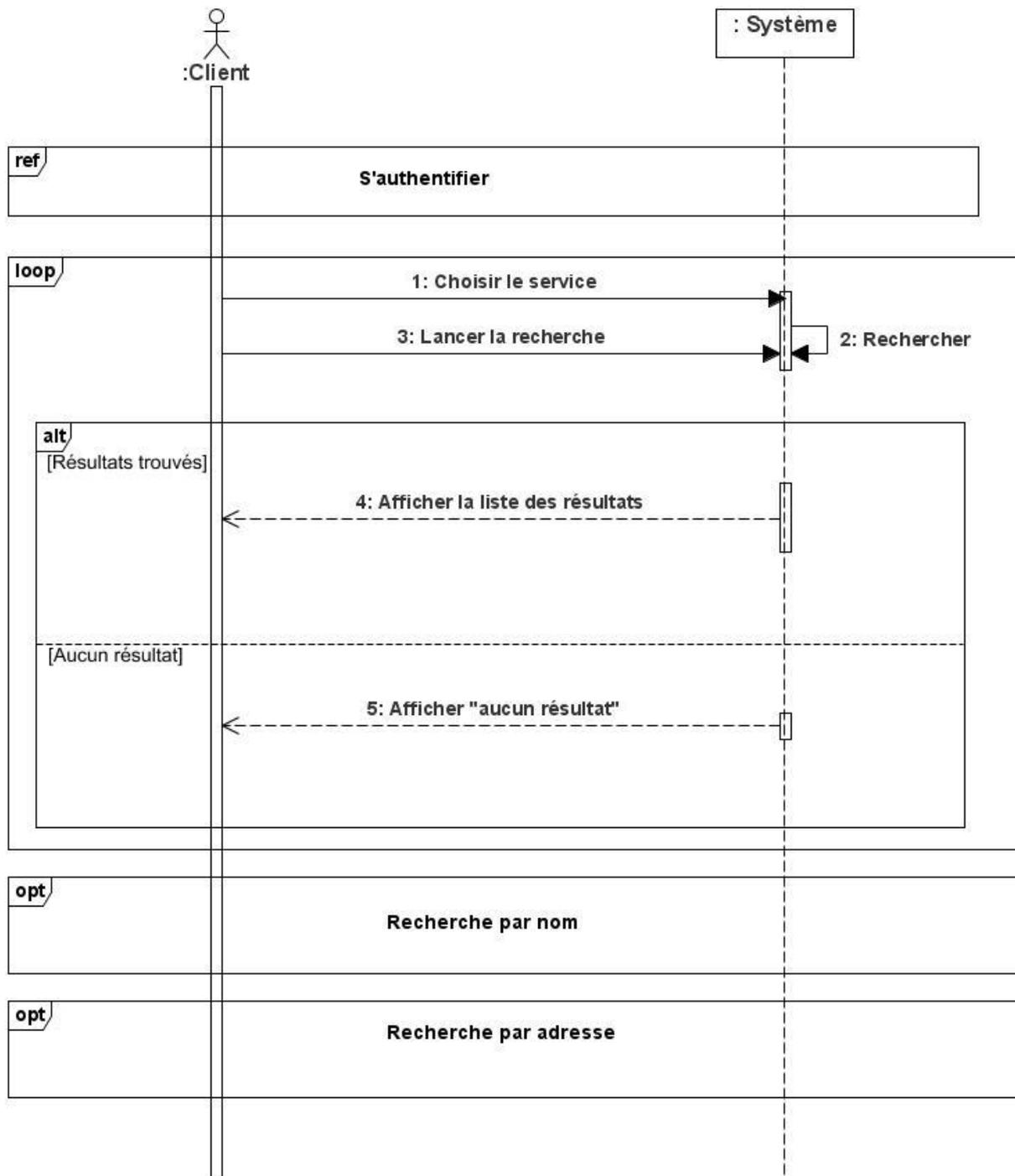


Figure 58 : Diagramme de séquences boîte noire du cas d'utilisation ‘rechercher un fournisseur par service’

8.6. Diagramme de séquences du cas d'utilisation “visualiser les informations d’un fournisseur”

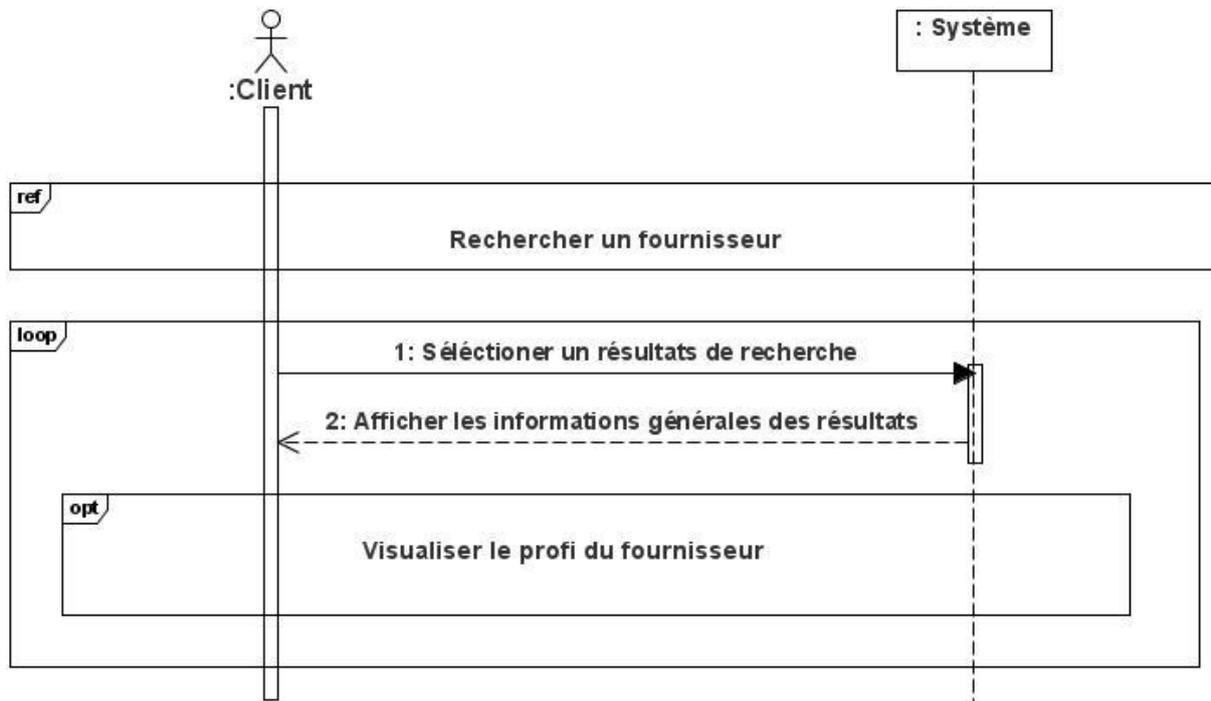


Figure 59 : Diagramme de séquence boîte noire du cas d'utilisation “visualiser les informations d’un fournisseur”

8.7 Diagramme de séquences du cas d'utilisation “afficher les détails sur un fournisseur”

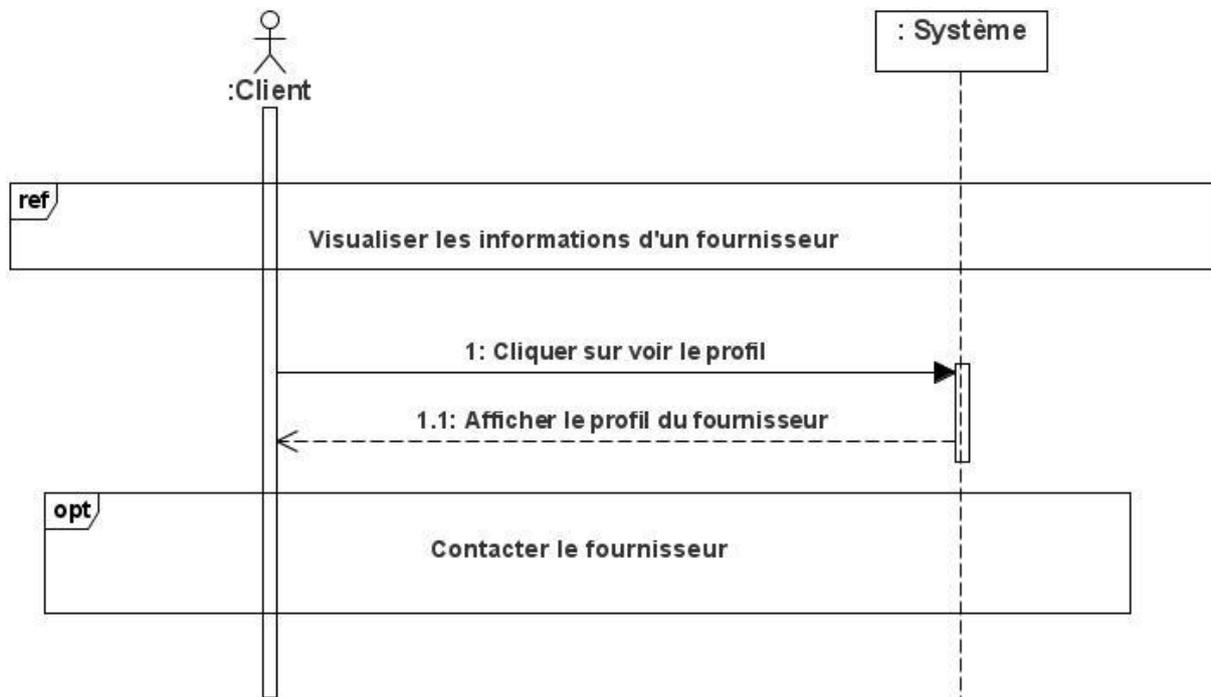


Figure 60 : Diagramme de séquences boîte noire du cas d'utilisation “afficher les détails sur un fournisseur”

8.8. Diagramme de séquences du cas d'utilisation "modifier le compte"

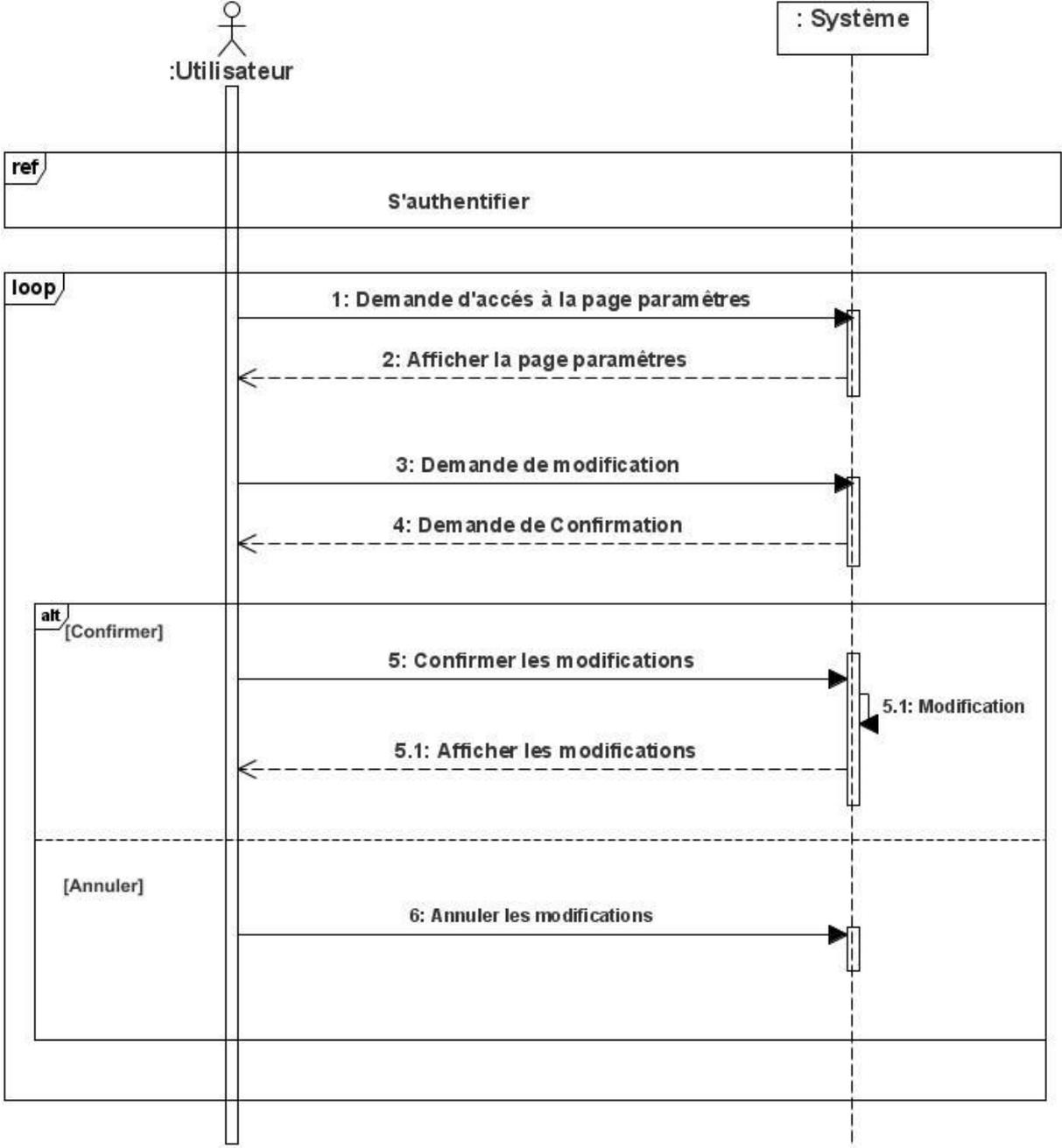


Figure 61 : Diagramme de séquences boîte noire du cas d'utilisation "modifier le compte"

8.9. Diagramme de séquences du cas d'utilisation ‘supprimer le compte’

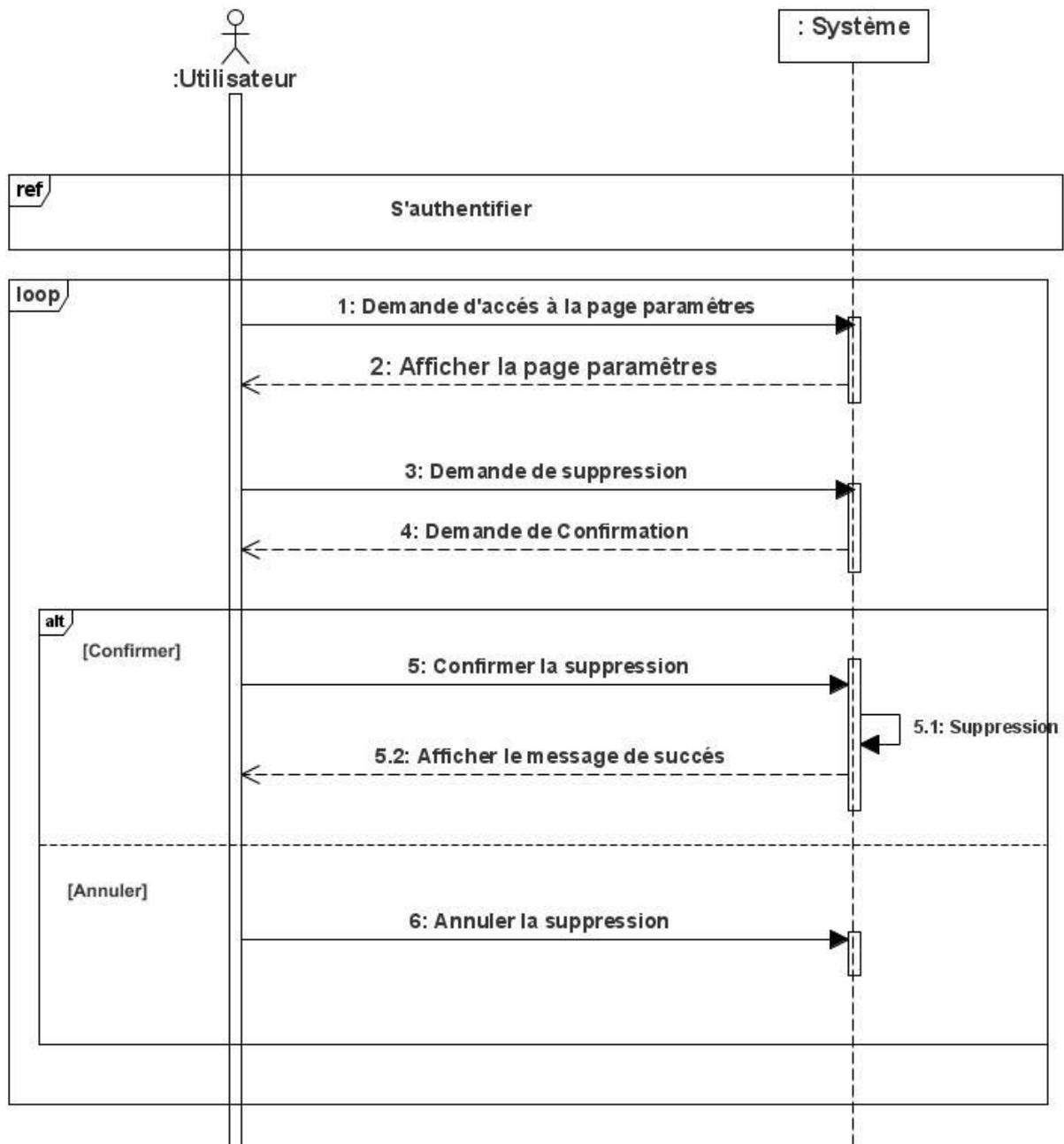


Figure 62 : Diagramme de séquences boîte noire du cas d'utilisation ‘supprimer le compte’

8.10. Diagramme de séquences du cas d'utilisation “se déconnecter”

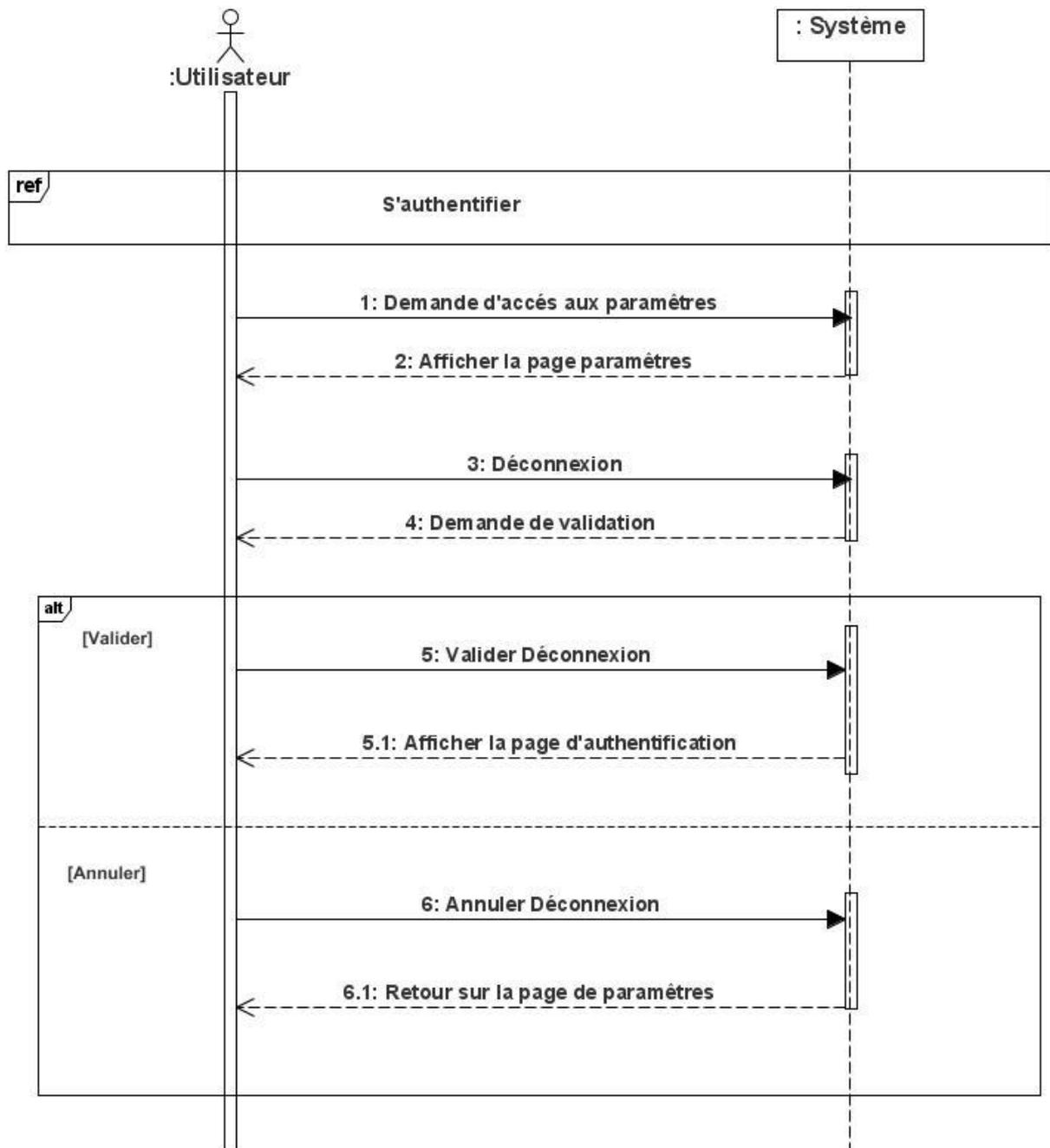


Figure 63 : Diagramme de séquences boîte noire du cas d'utilisation “se déconnecter”

9. Diagramme de classes d'analyse

Nous avons utilisé un diagramme de classe pour représenter le modèle du domaine qui n'est pas une description d'objets logiciels mais une visualisation des concepts du monde réel. On parle d'objets conceptuels. La figure 64 illustre ce diagramme.

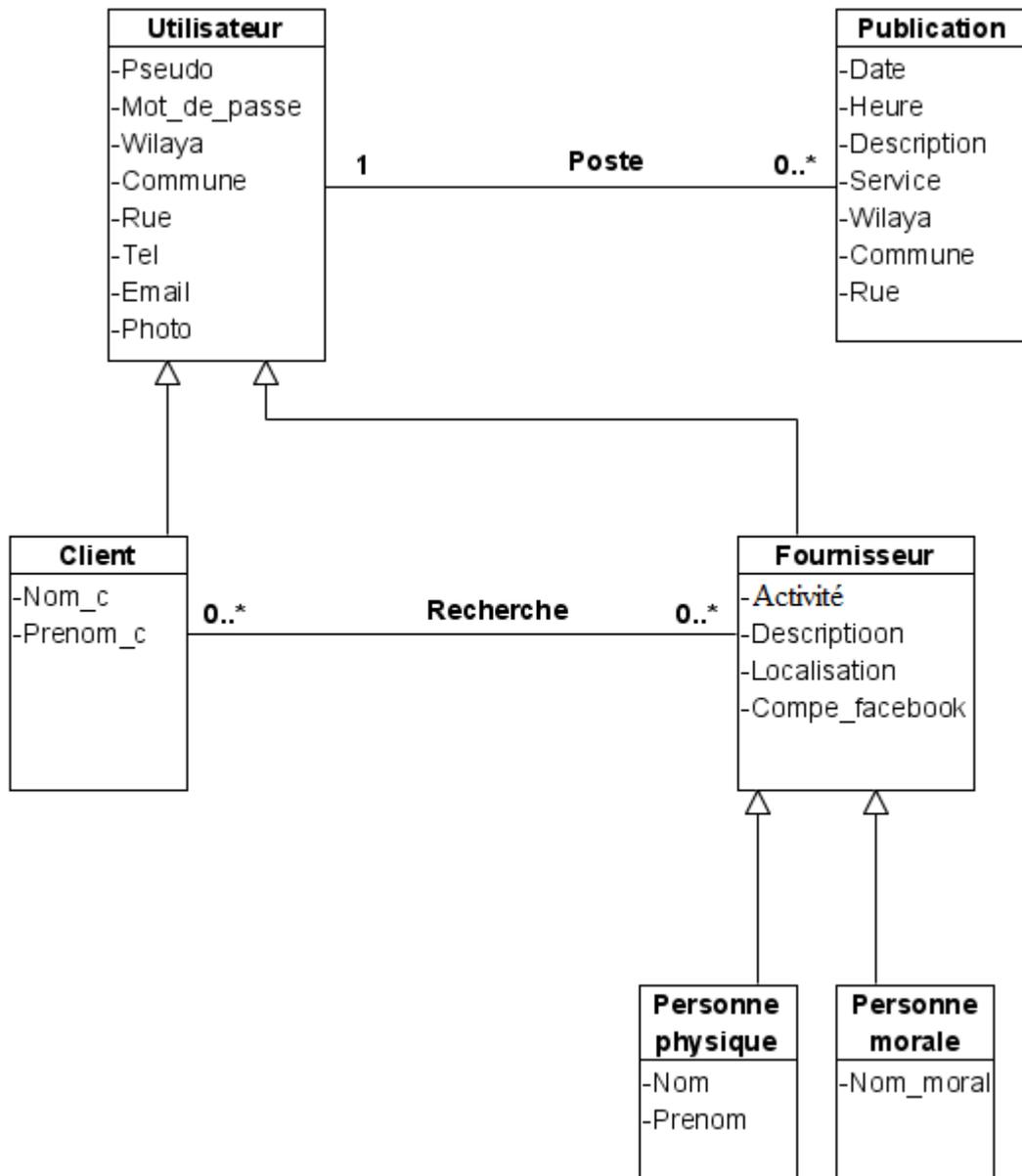


Figure 64 : Diagramme de classes de domaine

Conclusion

Au cours de ce chapitre nous avons détaillé dans sa première partie la démarche adoptée afin de réaliser notre projet.

Dans sa deuxième partie nous avons identifié les besoins de l'utilisateur ce qui nous a permis un découpage fonctionnel de la future application "ypiresia" à l'aide du diagramme des cas d'utilisation que nous avons détaillé, ensuite nous avons présentés des maquettes pour chaque cas d'utilisation et après nous avons illustré par des diagrammes de séquences. Pour conclure la phase d'analyse nous avons réalisé un diagramme de classes.

L'analyse réalisée dans ce chapitre va permettre d'implémenter et de réaliser complètement l'application "ypiresia", ce qui sera l'objet du prochain chapitre.

CHAPITRE 4 :
Réalisation
et évaluation

Introduction

Cette partie constitue le dernier volet de ce rapport. Après avoir terminé la phase d'analyse et conception, la solution étant déjà choisie et étudiée, il nous reste que de se décider comment l'implémenter sur l'environnement Windev mobile, exposer les choix techniques utilisés et le langage adopté, et présenter les interfaces et les tests réalisés.

1. Environnement matériel

Pour réaliser notre projet nous avons utilisé le matériel suivant :

- 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.

Nous avons utilisé un PC pour le développement de notre application « Ypiresia » et pour tester toutes ses fonctionnalités.

- 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.

Nous avons utilisé une tablette et un Smartphone pour tester toutes les fonctionnalités de notre application et surtout de tester les ancrages et les agencements des IHM.

2. La structure des fichiers de l'IHM

Pour développer notre application nous avons utilisés plusieurs fichiers comme :

- des fenêtres simples,
- des fenêtres internes,
- modèle de fenêtres,
- modèle des champs,
- des images,
- tests

Nous allons présenter les structures des fichiers que nous avons utilisés pendant notre développement :

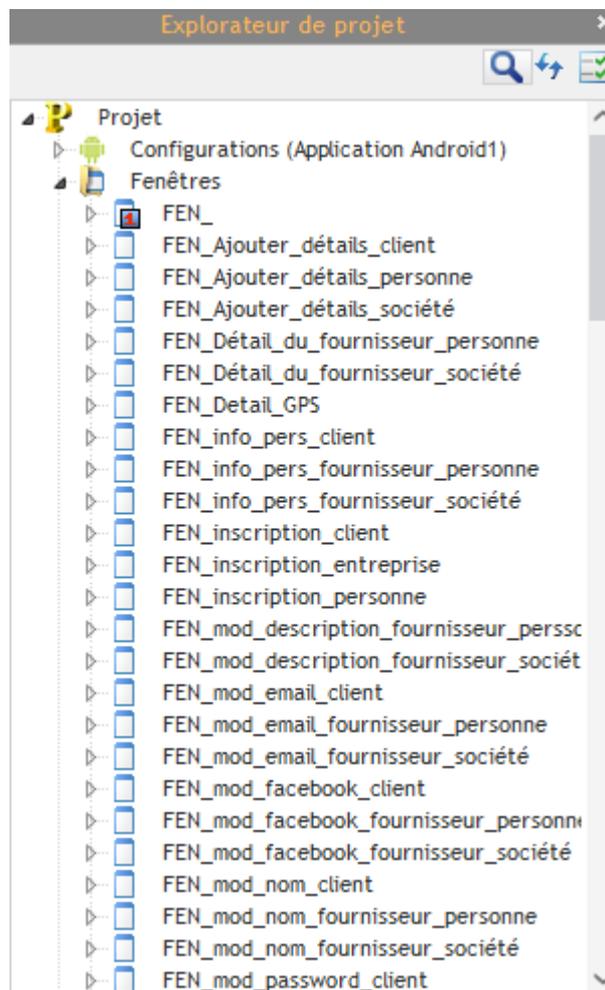


Figure 65 : Structure des fenêtres simples.

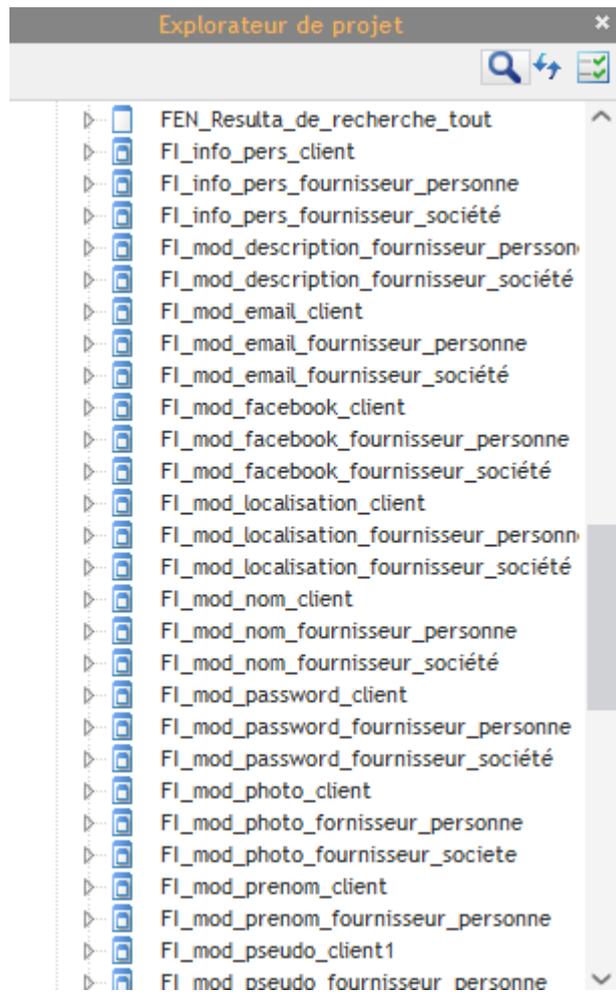


Figure 66 : Structure des fenêtres internes

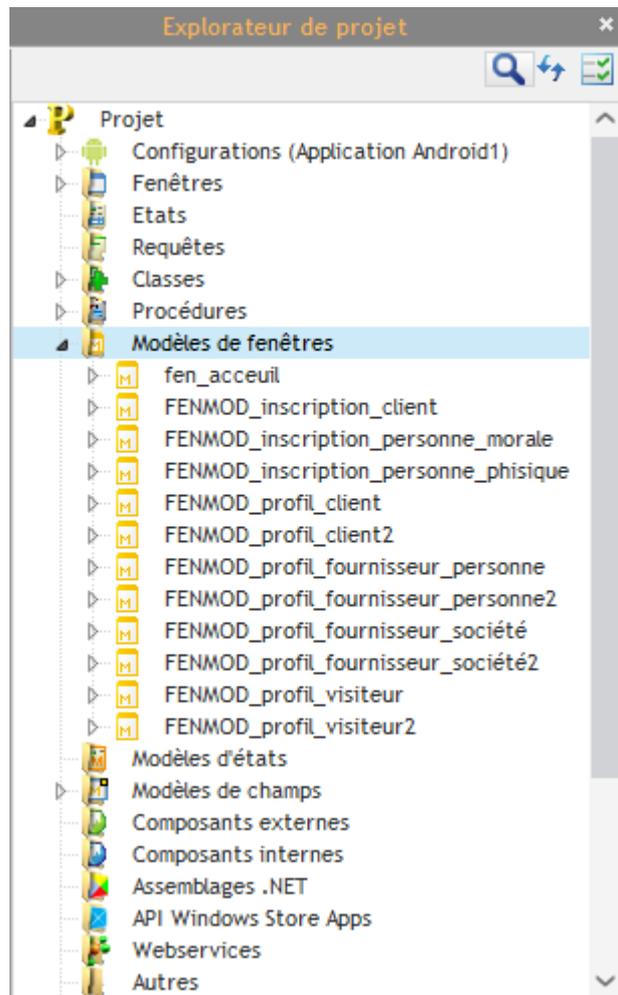
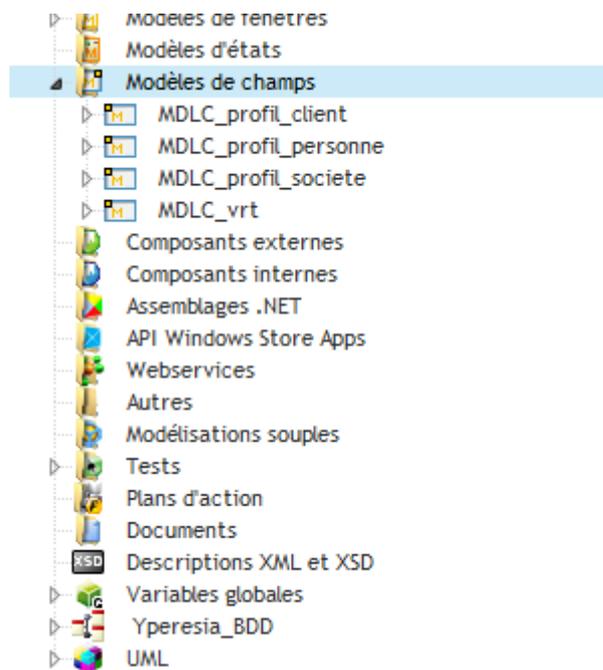
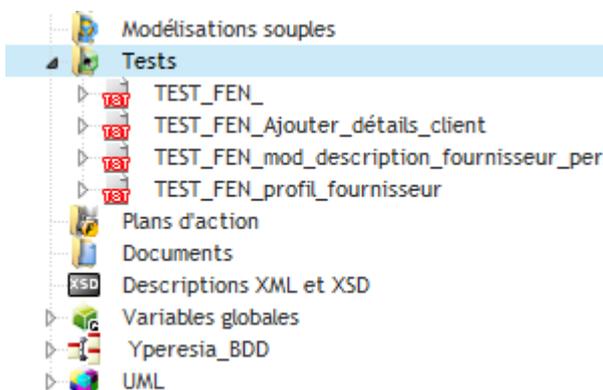


Figure 67 : Structure des modèles de fenêtres



Fenêtre 68 : Structure modèles de champs



Fenêtre 69 : Structure des tests

2.1 Format des fichiers de l'IHM

Les extensions des fichiers que nous avons utilisés sont comme suite :

Fichiers	Extensions
Fenêtres simples	.WPW
Fenêtres internes	.WDW
Modèle des fenêtres	.WDT
Modèle des champs	.WDT
Images	.PNG, .GIF
Tests de fenêtres	.WXT

Tableau 4 : Tableau des extensions de l'IHM

3. Capture d'écran relative aux différentes interfaces de l'application

Nous sélectionnons quelques fenêtres qui nous semblent importantes pour les intégrer dans le présent rapport :



Figure 70 : Fenêtre d'authentification en mode portrait

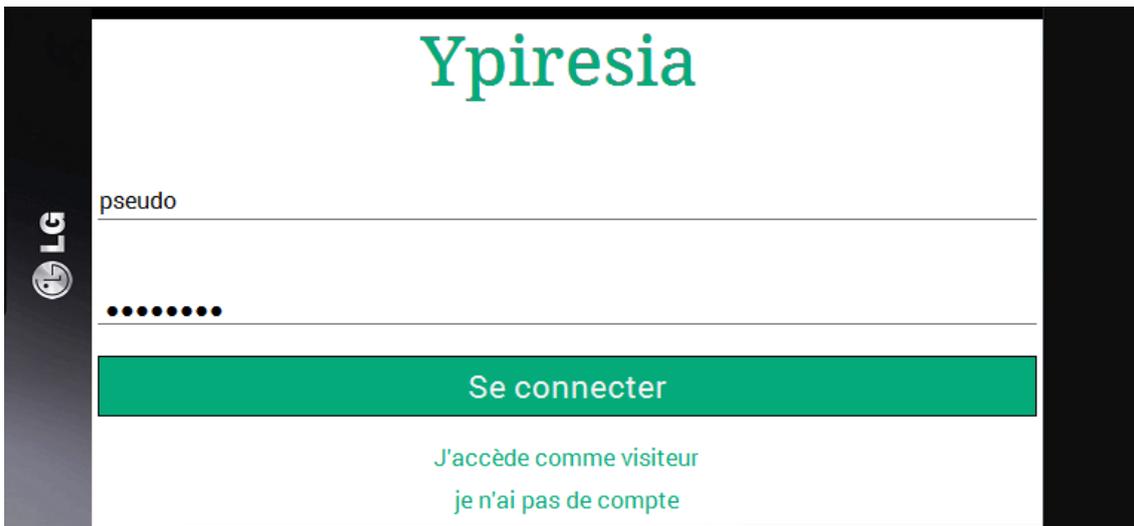


Figure 71 : Fenêtre d'authentification en mode paysage.



Figure 72 : Fenêtre d'inscription client en mode portrait

Inscription

Je suis Client ▼

Pseudo _____ Téléphone _____

Nom _____ E-mail _____

Prenom _____ Wilaya _____ ▼

Mot de passe _____ Commune _____

Confirmation _____ Rue _____

Valider

Figure 73 : Fenêtre d'inscription client en mode paysage

Inscription

Je suis personne physique ▼

Pseudo _____

Nom _____

Prenom _____

Mot de passe _____

Confirmation _____

Téléphone _____

E-mail _____

Activité _____ ▼

Wilaya _____ ▼

Commune _____

Rue _____

Valider

Figure 74 : Fenêtre d'inscription personne physique en mode portrait

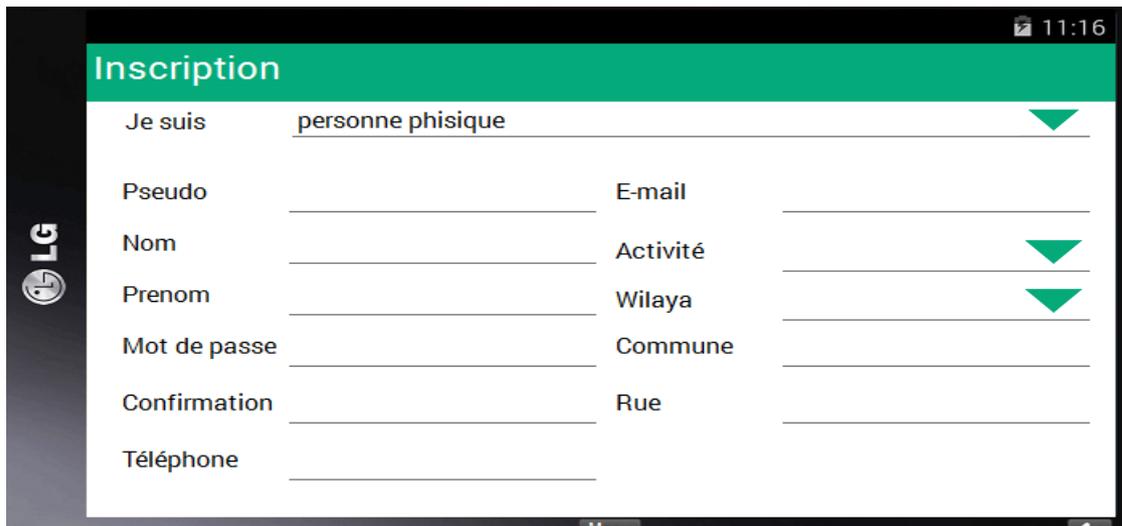


Figure 75 : Fenêtre d'inscription personne physique en mode paysage

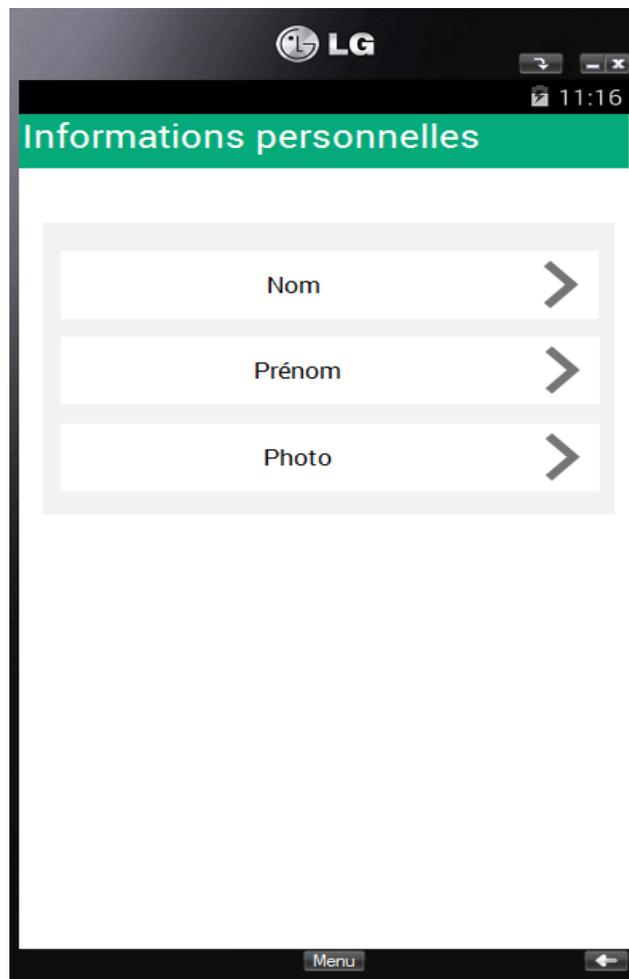


Figure 76 : Fenêtre Modification d'informations personnelles en mode portrait

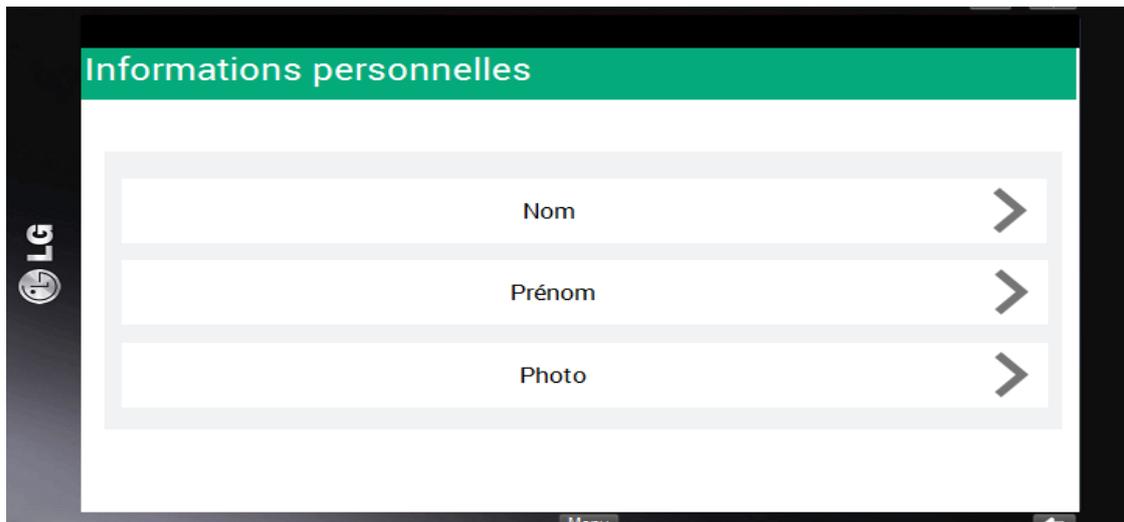


Figure 77 : Fenêtre Modification d'informations personnelles en mode paysage

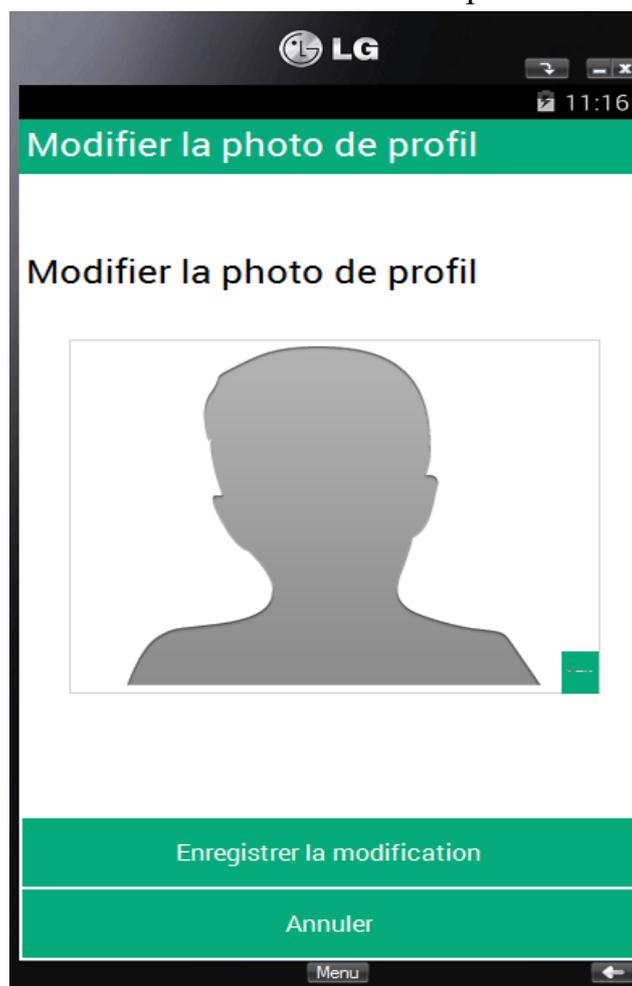


Figure 78 : Fenêtre Modifier la photo de profil d'un client en mode portrait

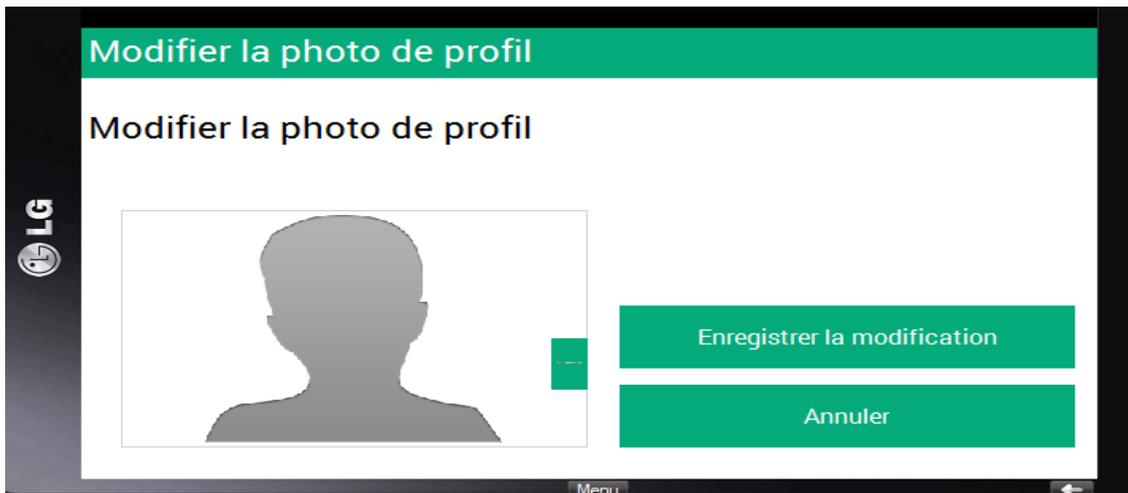


Figure 79 : Fenêtre Modifier la photo de profil d'un client en mode paysage

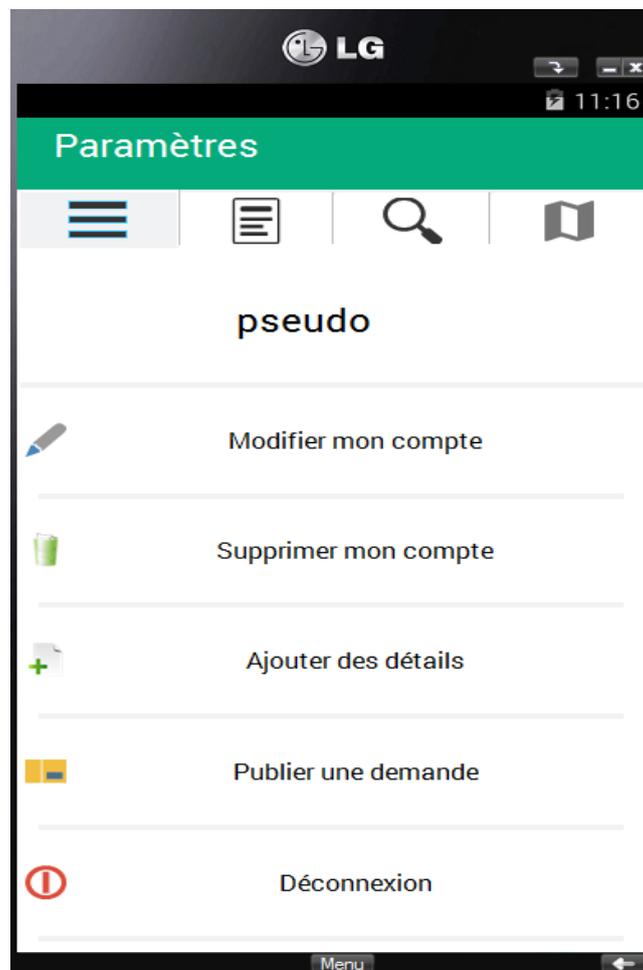


Figure 80 : Fenêtre correspondante à la rubrique Paramètres du profil client en mode portrait

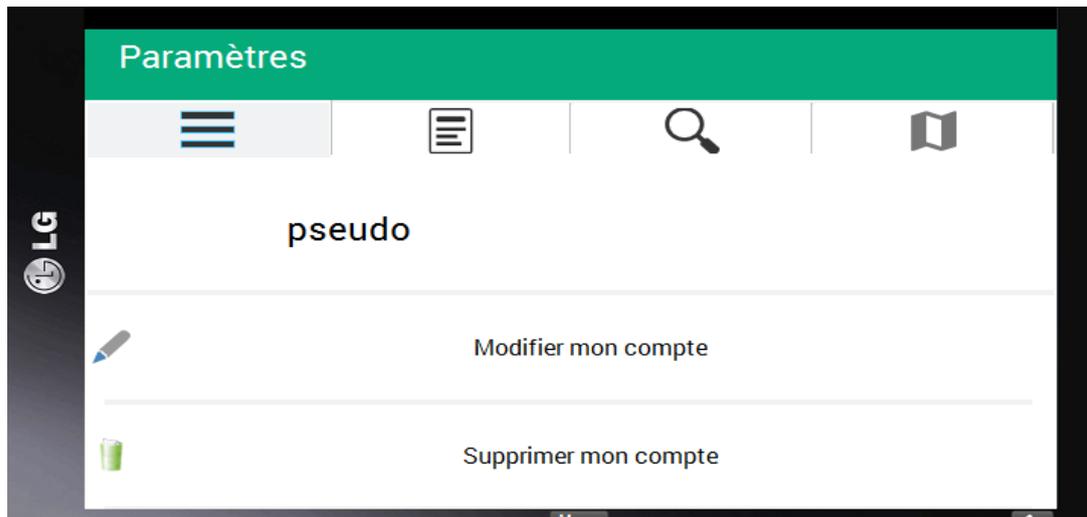


Figure 81 : Fenêtre correspondante à la rubrique Paramètres du profil client en mode paysage

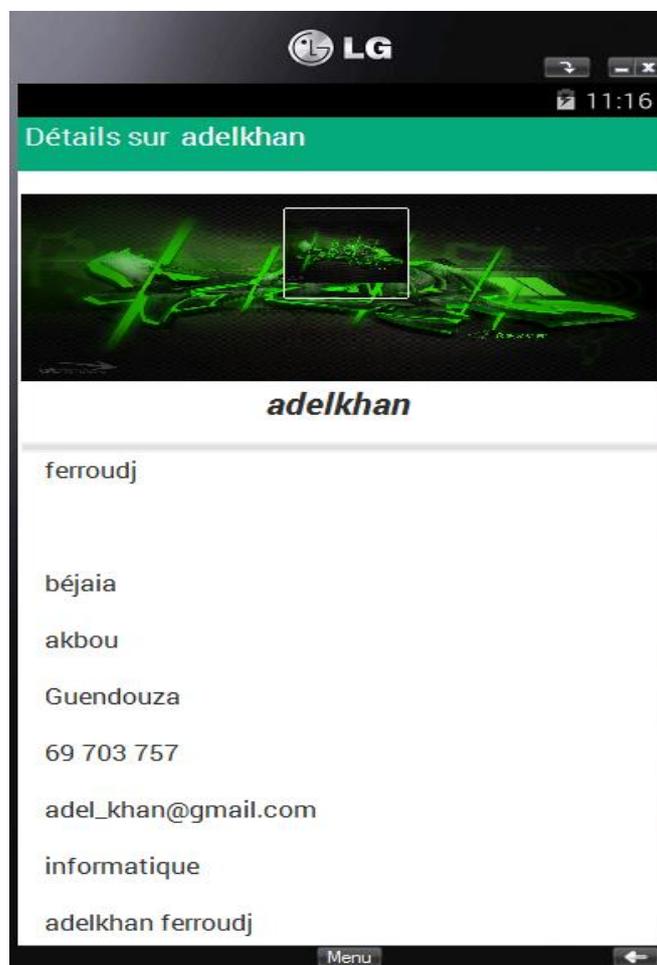


Figure 82 : Fenêtre Détails Résultats de la recherche

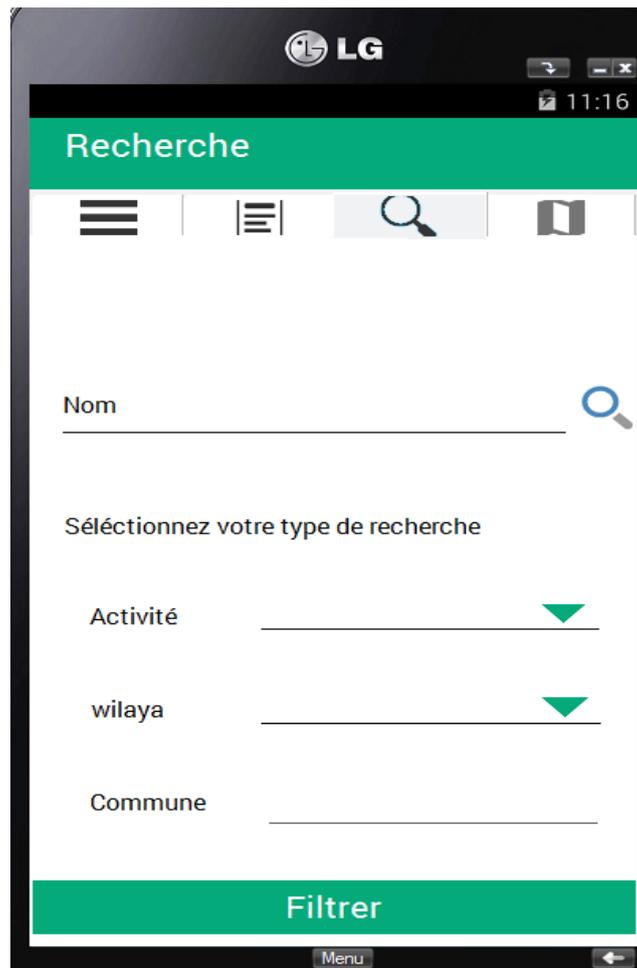


Figure 83 : Fenêtre correspondante à la rubrique Recherche du profil client en mode portrait

4. Choix de l'environnement logiciel

Pour développer l'application « Ypiresia » avec WinDev Mobile, nous avons installé :

Le JDK d'Oracle : Le JDK (Java Development Kit) distribué par Oracle est utilisé pour compiler les fichiers Java générés.

- Le SDK Android de Google : Le SDK Android (Software Development Kit ou Kit de développement logiciel) est un ensemble de fichiers et d'applications distribué par Google afin de permettre la compilation de notre application pour le système d'exploitation Android.
- Sur certains appareils, nous avons installé des fichiers complémentaires pour que le périphérique soit correctement détecté par le SDK et WinDev Mobile.
- Lors de la création d'un projet ou d'une configuration de projet pour Android, on a spécifié le chemin du SDK Android. [13]

5. Test de l'application

5.1 Test du projet

Il est possible de tester une application dans son ensemble, même si son développement n'est pas terminé. Dès qu'un problème apparaît dans l'exécution du projet, nous pouvons lancer le débogueur pour connaître et solutionner le problème rencontré.

Nous avons testé l'ensemble de notre projet en parallèle avec le développement ce qui nous a permis de simuler le lancement de l'application « Ypiresia » par l'exécutable.

Ce test est réalisé par deux moyens :

- Par un simulateur « Optimus G » de type Smartphone ou bien un simulateur « Sony Tablet S » de type tablette
- Par un émulateur Android livré par SDK.

5.2 Test d'une seule fenêtre

Dans notre application, nous pouvons tester une seule fenêtre, ceci permet d'exécuter uniquement la fenêtre en cours. Nous pouvons ainsi choisir de tester notre projet à partir d'une fenêtre donnée, ou de tester le fonctionnement d'une fenêtre dès que son développement est terminé. Comme pour le test du projet, nous pouvons aussi lancer le débogueur dès qu'un problème est rencontré.

La réalisation du test pour une fenêtre a été effectuée par le volet « Go » qui se trouve dans l'éditeur de fenêtre.

Au lancement du test, les traitements suivants sont exécutés :

- La déclaration des classes,
- L'initialisation du projet,
- La déclaration des globales de la fenêtre en cours,
- L'initialisation des champs présents dans la fenêtre en cours,
- L'initialisation de la fenêtre en cours,
- La prise de focus de la fenêtre en cours,
- L'entrée du premier champ de la fenêtre en cours (si ce champ a un traitement d'entrée).

5.3 Test d'IHM

Nous avons testé les ancrages et les agencements des IHM de notre application « Ypiresia », par l'utilisation d'un Smartphone de résolution (384*640) et une tablette de résolution (512*853).

Dans notre exemple une même fenêtre correspondante à modification de la photo de profil d'un client est représentée en deux miss en page différentes, qui est le principe de l'agencement (Figure 84).

Par contre, l'ancrage d'une même fenêtre de l'authentification d'un client représentée en deux terminaux différents avec deux résolutions différentes mais la position des éléments graphiques et le taux d'ancrage reste le même (Figure 85).

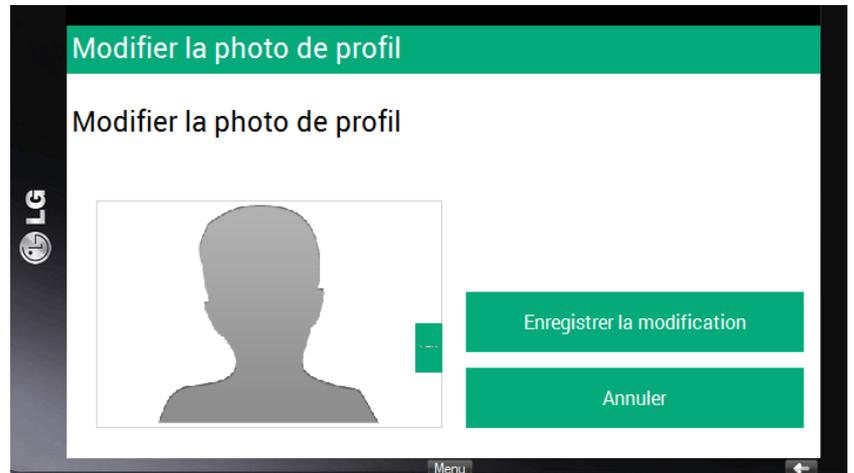


Figure 84 : Agencement de la fenêtre Modifier la photo de profil d'un client.

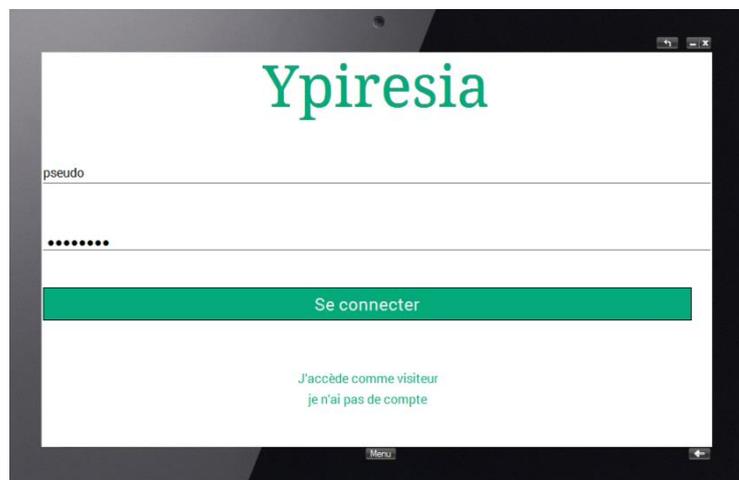


Figure 85 : Ancrage de la fenêtre d'authentification d'un client

6. Déploiement

L'étape finale de notre application « Ypiresia », appelée aussi 'phase de livraison' ou 'phase de mise en exploitation'. Dans cette étape nous avons regroupé toutes les activités et toutes les fonctionnalités qui mènent à l'installation et mise en marche de notre application. Nous avons installé notre application au format APK par :

- la génération de l'application sur le poste de développement.
 - la signature numérique du fichier APK ainsi obtenu. Note : une clé auto signée peut être utilisée.
 - la copie du fichier APK sur un appareil Android.
 - l'exécution du fichier APK directement sur l'appareil Android. Cette action provoque l'installation de l'application « Ypiresia ». Pour exécuter notre application, on a choisi son icône depuis le menu "Tous les programmes".
- [13]



Figure 86 : Génération de l'application

Conclusion

Dans cette dernière partie on a présenté l'environnement matériel et justifié les choix techniques pour la réalisation de l'application ypiresia, à savoir l'environnement logiciel. Ensuite on a exposé les interfaces graphiques et la structure physique du projet. En dernier, on a illustré le procédé de test et de déploiement.

Conclusion générale

Ces dernières années, la téléphonie mobile a été sans doute le secteur le plus dynamique, le plus rentable et le plus innovant de toute l'Industrie de l'informatique. Dans le cadre de notre projet, nous étions menées à explorer ce nouveau secteur et à développer une application mobile pour la mise en relation de services.

La principale difficulté rencontrée a clairement été le temps très limité dont nous avons disposé, surtout au vu du fait que nous étions confrontés à un nouvel environnement sur lequel nous n'avions aucune expérience. Toutefois, cela a pu être en partie compensé par le fait que la répartition du travail était facile et que nous avons pu accomplir notre travail qui consistait au prototypage et analyse des besoins liés au contexte de notre application.

Globalement, nous sommes satisfaits de l'avancement du projet, et même si nous ne remplissons pas toutes les fonctionnalités possibles, les objectifs établis au début du projet ont pour nous été accomplis.

Pour finir, ce mémoire nous conforte dans notre choix professionnel, et nous a permis d'appliquer nos connaissances acquises tout au long de notre parcours à l'université et en grande partie en génie logiciel.

Perspectives

L'étude que nous avons menée et l'application que nous avons développée sont le fruit d'un travail limité en temps et en expérience, c'est une œuvre imparfaite, nous en sommes conscients, mais il a contribué à nous rassembler et à appliquer notre savoir. Il est bien évidemment à enrichir et à améliorer.

Nous pensons déjà à enrichir l'application avec plus de fonctionnalités, améliorer les interfaces graphiques et l'ergonomie et sécuriser la gestion et le stockage des données.

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 multi utilisateur 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.

-Smaltalk :

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 à 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 tvOS

-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).

-CocoaTouch :

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 multi-plateforme 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 Crockford 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.

-APK :Android application package.

Fichier qui sert à distribuer et installer des logiciels d'application dans le système Android .

- SDK :Software Development Kit.

C'est un ensemble d'outils d'aide à la programmation pour concevoir des logiciels, applications mobile...etc., pour un matériel ou un système d'exploitation spécifique.

-JDK :Java Development Kit.

Un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en byte code destiné à la machine virtuelle Java.

liste des abréviations

- RIM** : Research In Motion.
- PDA**: Personal Digital Assistant.
- OS** : Operating System.
- HTML** : Hyper Text Markup Langage.
- CSS** : Cascading Style Sheets.
- API** : Application Programming Interface.
- PDF** : Portable Document Format.
- PHP**: Personal Home Page.
- SOAP** : Simple object Access Protocol.
- XML** : extensible Markup Langage.
- JSON** : JavaScript Object Notation.
- UML** : Unified Modeling Langage.
- IHM** : Interface Homme Machine.
- WCI** : WinDev Composants Internes.
- SMS** : Short Message System.
- WDA** : WinDev Analyse.
- ODBC** : Open database Connectivity.
- SQL** : Structured Query Langage.
- HFSQL** : Hyper File Structured Query Langage.
- WDD** : WinDev Donnés.

- POO**: Programmation Orienté Objet.
- SDK** : Software Developement Kit.
- JDK** : Java Developement Kit.
- APK** : Android application package.
- **SI** : Système d'information.

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 19/05/2016.

[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 21/05/2016.

[4]:Majdi belaghgi. Cours universitaire. Génie logiciel.2006. Repéré à: <http://www.isetjb.rnu.tn/docs/divers/ouvrages-iset-djerba-2015%20.pdf>. Consulté le 29/05/2016.

[5]:Les environnements de développement [Article].Repéré à: <http://general.developpez.com/edi.html> Consulté le 05/06/2016.

[6]:Le développement logiciel.[Article]. Repéré à: <http://www.blog.erlem.fr/programmation/edi.html> Consulté le 05/06/2016.

[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 12/06/2016.

[8]: PC SOFT. E-book Windev mobile 21, concepts. Repéré à:
<http://www.pcsoft.fr/windevmobile/concepts.html> Consulté le 12/06/2016.

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

Consulté le 13/06/2016.

[10]:PC SOFT. Documentation officielle de Windev mobile. Repéré à:
<http://doc.pcsoft.fr> Consulté le 19/06/2016.

[11]:e-book meilleur cours de windev. Repéré à:
<http://wind.developpez.com/tutoriels> Consulté le 21/06/2016.

[12]:Thierry Baccino. Conception, design des documents numériques vol 2
.2009.Lavoisier.

[13]:Tutoriel windev mobile. Repéré à:
<http://www.cours-gratuit.com/cours-windev/cours-windev-complet-en-pdf>.
Consulté le 21/06/2016.

ANNEXE

A.1 Java: jdk-6u14-windows-i586

Il est indispensable pour qu'windev-java-indigo-SR2-win32-x86_64 marche sur PC

A.2 Java: jre-7u4-windows-x64

Il est indispensable pour qu'wiwndev-java-indigo-SR2-win32-x86_64 marche sur PC

A.3 Installation du SDK Android

Pour développer des applications Android avec WinDev Mobile, il est nécessaire d'installer :

- Le JDK de Oracle
- Le SDK Android de Google : Le SDK Android (Software Development Kit ou Kit de développement logiciel)
- Sur certains appareils, il est nécessaire d'installer des fichiers complémentaires

Lors de la création d'un projet ou d'une configuration de projet pour Android, il est nécessaire de spécifier le chemin du SDK Android.

Le SDK Android est disponible en téléchargement sur le site de Google consacré au développement pour Android (<http://developer.android.com>). Pour développer des applications Android avec WinDev Mobile, seul le "Stand-alone SDK " est nécessaire.

La version pour Windows du SDK "Stand-alone" se trouve à l'adresse suivante : <http://developer.android.com/sdk/installing/index.html?pkg=tools>.

- Cliquez sur "Download the SDK Now".
- Téléchargez le SDK pour Windows.

Ces adresses sont valides à l'heure de l'écriture de cette documentation.

Comment installer l'Android ?

La procédure à suivre est la suivante :

1. Décompressez le fichier téléchargé dans un dossier à la racine du lecteur "C:" pour éviter une virtualisation de Windows. Par exemple : "C:\Android\".
2. Ouvrez le dossier avec l'explorateur de Windows.
3. Lancez le programme "SDKManager.exe". Ce programme va permettre l'installation des différentes versions existantes du SDKAndroid ainsi que leurs mises à jour.

Il peut être nécessaire d'ajouter le chemin du sous-répertoire "Tools" dans le PATH de votre poste. Pour cela :

- Affichez les propriétés du poste de travail.
- Dans l'onglet "Avancé", cliquez sur le bouton "Variables d'environnement".
- Sélectionnez l'option "PATH" dans les variables système et cliquez sur "Modifier".
- Ajoutez le chemin complet du répertoire "Tools".

Le programme "SDKManager.exe" peut afficher le message d'erreur suivant :
"Failed to fetch URL https://dl-ssl.google.com/android/repository/repository.xml, reason: HTTPS SSL error. You might want to force download through HTTP in the settings."

Ce message indique que le gestionnaire de SDK Android n'a pas réussi à établir une connexion HTTPS pour télécharger les composants du SDK. Pour résoudre ce problème, il faut :

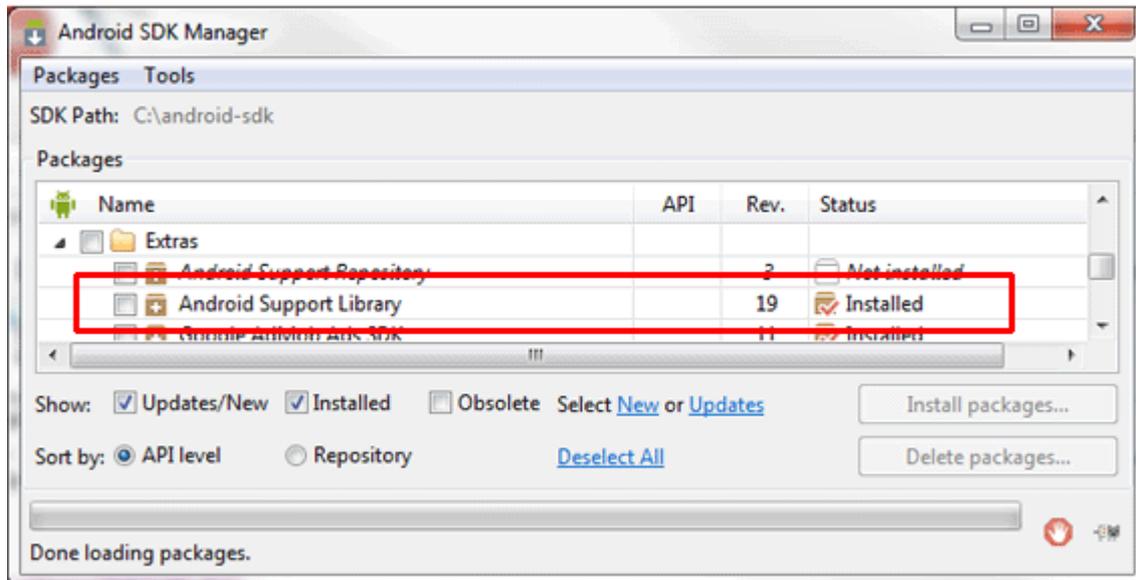
- Cliquer sur "Settings" dans la liste à gauche de la fenêtre.
- Cocher la case "Force https://... source to be fetched using http://".
- Cliquer "Save & Apply".
- Le chemin du répertoire de décompactage devra être indiqué dans l'assistant de création de projet de WinDev Mobile.

Version du SDK Android nécessaire pour WinDev Mobile

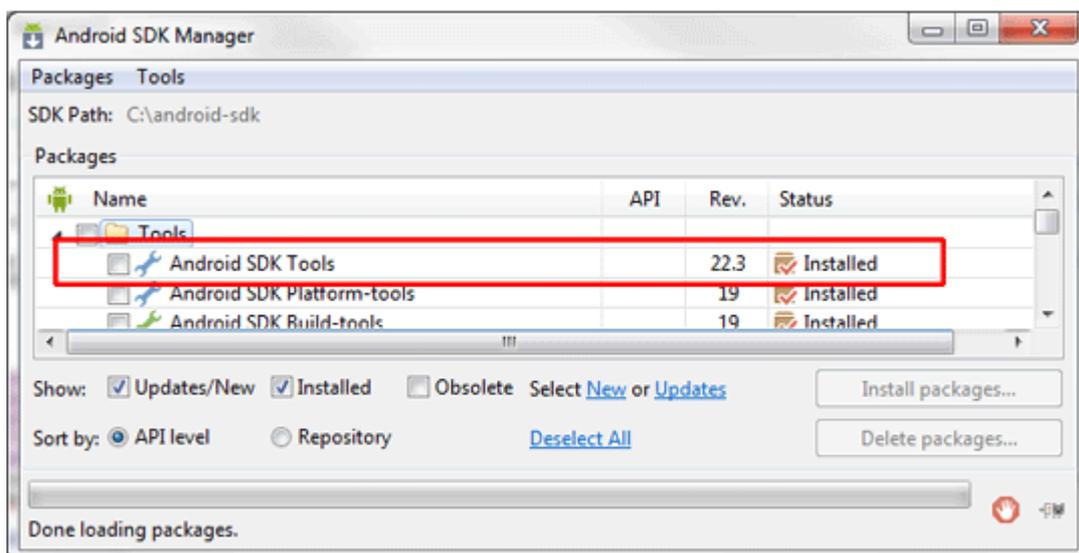
WinDev Mobile a besoin au minimum de la version 1.5 du SDK Android. Cette version permet le développement d'applications pour Android 1.5 et supérieur. Si une version plus récente est installée (par exemple la 2.0), WinDev Mobile utilisera cette version. Les applications générées resteront compatibles avec la version 1.5 de Android.

Pour profiter de la détection de la plateforme Android depuis WinDev Mobile, il est également conseillé d'installer le package "USB Driver".

A partir de la version 19, pour générer une application Android, il est nécessaire d'avoir installé une librairie "Android Support Library" supérieure à la version 17 .



Pour que la compilation optimisée soit activée, il est nécessaire d'installé une version du SDK-tool en version du sdk-tool en version 21 ou supérieure.



Impact de l'installation du SDK Android 2.3 API9 (version r08)

L'installation de la version 2.3 (Gingerbread) du SDKAndroid n'a plus la même structure que les SDK précédents. WinDev Mobile 16 et supérieur supporte cette version du SDK.

Cas particulier : WinDev Mobile 15

WinDev Mobile 15 ne supporte pas cette version du SDK. Si la version 2.3 du SDK a été installée sur le poste de développement, il est nécessaire d'effectuer certaines opérations dans le répertoire d'installation du SDK pour pouvoir compiler et/ou exécuter les applications Android générées par WinDev Mobile (quelle que soit la version du SDK utilisée pour compiler).

Pour déployer les applications sur l'émulateur ou le téléphone (quelle que soit la version du SDK utilisée pour générer l'application) : copier les fichiers Adb.exe, AdbWinApi.dll et AdbWinUsbApi.dll situés dans le répertoire <répertoire SDK>/platform-tools vers le répertoire <répertoire SDK>/tools.

Pour générer et déployer les applications avec la version 2.3 du SDK:

- copier les fichiers Adb.exe, AdbWinApi.dll et AdbWinUsbApi.dll situés dans le répertoire <répertoire SDK>/platform-tools vers le répertoire <répertoire SDK>/tools.
- créer le répertoire <répertoire SDK>/platforms/android-9/tools
- copier le répertoire <répertoire SDK>/platform-tools/lib dans <répertoire SDK>/platforms/android-9/tools
- copier les fichiers aapt.exe, aidl.exe, dexdump.exe et dx.bat situés dans le répertoire <répertoire SDK>/platform-tools vers le répertoire <répertoire SDK>/platforms/android-9/tools

Systèmes d'exploitation supportés par leSDKAndroid

Le SDK Android fonctionne sur les versions suivantes de Windows :

- Windows XP (en 32 bits uniquement)
- Windows Vista et versions supérieures (en 32 et 64 bits)

A.4 JDK de Oracle utilisé pour le développement Android

Où trouver le JDK ?

Le JDK est disponible en téléchargement sur le site de Oracle à l'adresse suivante :

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

Le JDK seul est suffisant. Il n'est pas nécessaire de télécharger les versions packagées avec des modules complémentaires (JavaFX ou NetBeans par exemple).

Version du JDK nécessaire pour WinDev Mobile

WinDev Mobile a besoin au minimum de la version 5 du JDK.

Si une version plus récente est installée (par exemple la version 6), WinDev Mobile utilisera cette version.

- Le JDK 5 est parfois appelé JDK 1.5. Ces deux dénominations désignent le même produit.
- L'installation d'une version plus récente du JDK n'a pas d'incidence sur la génération d'applications Java avec WinDev.
Rappel : Pour générer des applications Java avec WinDev, le JDK 1.4.2 est nécessaire au minimum.
- Il est nécessaire d'installer la version 32 bits du JDK, y compris sur un système d'exploitation 64 bits.

Systèmes d'exploitation supportés par le JDK

Le JDK pour Windows fonctionne sur toutes les versions de Windows à partir de Windows 2000.