

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abd elhafid boussouf Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatique

**Mémoire préparé En vue de l'obtention du diplôme de
Licence
En : Filière informatique générale**

**Thème : développement d'un site web
pour une agence immobilière**

**Préparé par : Mokhnache Abdelhak
Khenfri Abdelhakim
Bouhali Zakarya**

Encadrer par : Mlle khalfi souheila

Année universitaire : 2014/2015

Remerciements

Louange à dieu tout puissant de nous avoir aidé et éclairé le chemin pour achever notre travail et nos études.

Nos remerciements à nos très chers parents, frères, sœurs, collègues et amis respectives qui nous ont encouragés et soutenu durant tout notre parcours.

Un remerciement particulier à notre encadreur **khalfi souhila** pour sa présence, son aide et surtout pour ses précieux conseils qui nous ont assistés pour l'accomplissement de notre projet malgré les préoccupations administratives.

Nous tenons à exprimer nos sincères remerciements à tout le personnel de l'institut de l'informatique surtout les enseignants qui nous ont enseigné durant toutes nos années d'étude. Enfin nous remercions toutes personnes qui ont contribué de près ou de loin à l'achèvement de ce travail.

Merci bien.

Sommaire

TNTRODUCTION GENERALE.....	1
CHAPITREI : TECHNOLOGIE WEB	
1.1 Introduction.....	5
1.2 Internet.....	5
1.2.1 Qu'est ce qu'internet.....	5
1.2.2 Histoire d'Internet.....	6
1.3 Le Web.....	6
1.3.1 World Wide Web (WWW).....	6
1.3.2 Site Web.....	7
1.3.2.1 Quelques définitions.....	7
1.3.2.2 Les types de site web.....	8
1.3.2.2.1 Les sites web statiques.....	8
1.3.2.2.2 Les sites web dynamiques.....	9
1.3.2.2.3 Quel type de site pour quelle utilisation ?	10
1.4 Conclusion.....	10
CHAPITREII : UML	
2.1 Introduction.....	12
2.2 Présentation du langage de modélisation UML.....	12
2.2.1 Historique d'UML.....	12
2.2.2 Définition d'UML.....	12
2.2.3 Pourquoi UML ?.....	13

2.2.4 Présentation des vues et diagrammes d'UML.....	13
2.2.4.1 la vue statique	14
2.2.4.2 la vue dynamique	14
2.2.5 Description de quelques diagrammes UML.....	14
2.2.5.1 Le diagramme de cas d'utilisation.....	14
1 Définition.....	14
2 Éléments constitutifs des cas d'utilisation.....	14
2.2.5.2 Le diagramme de classe.....	16
1 Définition	16
2 Éléments constitutifs de diagramme de classe.....	17
2.2.5.3 Diagramme de séquence.....	18
1 Définition	18
2 Éléments constitutifs de diagramme de séquence.....	18
2.2.5.4 Diagramme d'activités	20
1 Définition	20
2 les éléments constitutifs.....	20
2.2.6 Avantages et inconvénients d'UML.....	21
2.2.6.1 Les points forts d'UML.....	21
2.2.6.2 Les points faibles d'UML.....	21
2.3 Démarche de modélisation.....	22
2.3.1 Identification des besoins et spécification des fonctionnalités.....	22
2.3.1.1 Identification et représentation des besoins: Diagramme de cas d'utilisation.....	22

2.3.1.2	Spécification détaillée des besoins : diagrammes de séquence système.....	23
2.3.1.3	Maquette de l'IHM de l'application (non couvert par UML.....)	23
2.3.2	Phases d'analyse.....	23
2.3.2.1	Analyse du domaine (modèle du domaine).....	23
2.3.2.2	Diagramme de classes participantes.....	23
2.3.2.3	Diagrammes d'activités de navigation.....	24
2.3.3	Phase de conception.....	24
2.3.3.1	Diagrammes d'interaction.....	24
2.3.3.2	Diagramme de classes de conception.....	25
2.4	Conclusion.....	26

CHAPITRE IV : ETUDE DE CAS

3.1	Introduction.....	28
3.2	Identification des besoins.....	28
3.2.1	Diagramme de cas d'utilisation	28
3.2.1.1	Identification des acteurs.....	28
1.	L'administrateur.....	28
2.	Le propriétaire	28
3.	Le client	28
3.2.1.2	diagramme de cas d'utilisation.....	28
3.2.1.3	Description textuelle des cas d'utilisation	29
3.2.1.3.1	Description du Cas d'utilisation « authentification ».....	29
3.2.1.3.2	Description du cas d'utilisation « créer compte ».....	30
3.2.1.3.3	Description du Cas d'utilisation « proposer un bien ».....	30
3.2.1.3.4	Description du Cas d'utilisation « modifier un bien ».....	31

3.2.1.3.5 Description du Cas d'utilisation « supprimer un bien ».....	31
3.2.1.3.6 Description du Cas d'utilisation « recherche bien »	32
3.2.1.3.7 Description du cas d'utilisation « Consulter une annonce ».....	32
3.2.1.3.8 Description du cas d'utilisation « choisir une annonce ».....	33
3.2.1.3.9 Description du cas d'utilisation « gérer compte ».....	33
3.2.2 Les diagrammes de séquences système.....	34
3.2.2 .1 Diagramme de séquence Authentification.....	34
3.2.2 .2 Diagramme de séquence créer un compte.....	35
3.2.2 .3 Diagramme de séquence Gérer un bien.....	36
3.2.2 .3.1 proposer un bien.....	36
3.2.2 .3.2 modifier un bien	37
3.2.2 .3.3 supprimer un bien	38
3.2.2 .4 Diagramme de séquence recherche un bien.....	39
3.2.2 .5 Diagramme de séquence consulter annonce.....	40
3.2.2 .6 Diagramme de séquence choisir annonce.....	41
3.2.2 .7 Diagramme de séquence Gérer compte.....	42
3.2.2.7.1 Supprimer un compte.....	42
3.2.2.7.2 Valider un compte.....	43
3.3 La phase d'analyse	44
3.3.1 : Modèle du domaine.....	44
3.3.2 : diagramme de class participantes.....	45
3.3.2 .1 Authentification.....	45
3.3.2 .2 proposer un bien.....	46
3.3.2 .3 modifier un bien.....	47
3.3.2 .4 supprimer un bien.....	48
3.3.2 .5 consulter annonce.....	49
3.3.2 .6 utiliser annonce	50

3.3.3 Diagrammes d'activités.....	51
3.3.3 .1 Authentification.....	51
3.3.3 .2 proposer un bien.....	52
3.3.3 .3 modifier un bien.....	53
3.3.3 .4 supprimer un bien.....	54
3.3.3 .5 consulter annonce	55
3.3.3 .6 choisir une annonce.....	56
3.4 Conclusion	57

CHAPITRE 4 : IMPLEMENTATION

4.1 Introduction.....	59
4.2 Les Langages de programmation.....	59
4.2.1 Le langage HTML.....	59
4.2.2 Java Script.....	59
4.2.3 Le langage PHP (Personnel Home Page).....	59
4.2.4 MYSQL.....	61
4.3 Les outils de développement.....	61
4.3.1 UML diagramme.....	61
4.3.2 PH PMY Admin.....	61
4.4 Le passage du diagramme de classe au modèle relationnel.....	63
1) Transformation des entités /classes.....	63
2) Transformation des associations.....	63
4.5 Présentation de quelques interfaces de l'application.....	65
4.5.1 page d'accueil	65
4.5.2 Nos annonces	65
4.5.3 Page d'inscrire.....	66
4.5.4 A propose	66

4.5.5 Log in.....	67
4.5.6 espace propriétaire	67
4.5.7 proposer un bien	68
4.5.8 Gérer un bien.....	68
4.5.9 supprimer un bien.....	69
4.5.10 modifier un bien.....	69
4.5.11 Choisir une annonce.....	70
4.5.12 gérer les comptes.....	70
4.6 Conclusion.....	71
Conclusion générale.....	72

Liste des figures

Figure 1.1 : interconnexion entre les ordinateurs.....	5
Figure 1.2 : Logo historique du World Wide Web par Robert Cailliau.....	7
Figure 1.3 : site statique.....	8
Figure 1.4 : site dynamique.....	9
Figure 2.1 : historique d'UML.....	14
Figure 2.2 : diagramme de cas d'utilisation.....	17
Figure 2.3 : représentation graphique d'une classe.....	18
Figure 2.4 : diagramme de classe.....	19
Figure 2.5 : de diagramme de séquence.....	21
Figure 2.6 : Représentation graphique d'un diagramme d'activité.....	22
Figure 2.7 : Quelle méthode pour passer de l'expression des besoins au code de l'application ?.....	23
Figure 2.8 : Chaîne complète de la démarche de modélisation du besoin jusqu'au code.....	24
Figure 3.1 : diagramme de cas d'utilisation.....	29
Figure3.2 : diagramme de séquence Authentification.....	34
Figure3.3 : diagramme de séquence créer un compte.....	35
Figure3.4 : diagramme de séquence proposer un bien.....	36
Figure3.5 : diagramme de séquence modifier un bien.....	37
Figure3.6 : diagramme de séquence supprimer un bien.....	38
Figure3.7 : diagramme de séquence recherche un bien.....	39
Figure3.8 : diagramme de séquence consulter annonce.....	40
Figure3.9 : diagramme de séquence choisir annonce.....	41

Figure3.9 : diagramme de séquence suppression d'un compte.....	42
Figure3.10 : diagramme de séquence valider d'un compte.....	43
Figure3.11 : diagramme de class.....	44
Figure3.12 : diagramme de class participant Authentification.....	45
Figure3.13 : diagramme de class participant proposer un bien.....	46
Figure3.14 : diagramme de class participant modifier un bien.....	47
Figure3.15 : diagramme de class participant supprimer un bien.....	48
Figure3.16 : diagramme de class participant consulter annonce.....	49
Figure3.17 : diagramme de class participant choisir annonce.....	50
Figure3.18 : diagramme d'activité Authentification.....	51
Figure3.19 : diagramme d'activité proposer un bien.....	52
Figure3.20 : diagramme d'activité modifier un bien.....	53
Figure3.21 : diagramme d'activité supprimer un bien.....	54
Figure3.22 : diagramme d'activité consulter annonce.....	55
Figure3.23 : diagramme d'activité choisir une annonce.....	56
Figure4.1 : UML diagramme version:6.13.....	62
Figure 4.2 : PHP MY Admin.....	63
Figure 4.3 : structure de base de données.....	64
Figure 4.4 : page d'accueil.....	65
Figure 4.5 : nos annonces.....	65
Figure 4.6 : page d'inscrire.....	66
Figure 4.7 : A propose.....	66
Figure 4.8 : log in.....	67
Figure 4.9 : Espace propriétaire.....	67
Figure 4.10 : proposer un bien.....	68
Figure 4.11 : Gérer un bien.....	68

Figure 4.12 : Supprimer un bien.....	69
Figure 4.13 : modifier un bien.....	69
Figure 4.14 : choisir une annonce.....	70
Figure 4.15 : Gérer les comptes.....	70

Liste des tableaux

Tableaux 3.1 : Description du Cas d'utilisation « authentification ».....	29
Tableaux 3.2 : Description du cas d'utilisation « créer compte ».....	30
Tableaux 3.3 : Description du Cas d'utilisation « proposer un bien ».....	30
Tableaux 3.4 : Description du Cas d'utilisation « modifier un bien ».....	31
Tableaux 3.5 : Description du Cas d'utilisation « supprimer un bien »... ..	31
Tableaux 3.6 : Description du Cas d'utilisation « recherche bien ».....	32
Tableaux 3.7 : Description du cas d'utilisation « consulter une annonce ».....	32
Tableaux 3.8 : Description du cas d'utilisation « choisir une annonce ».....	32
Tableaux 3.9 : Description du Cas d'utilisation « supprimer un compte ».....	33
Tableaux 3.9 : Description du Cas d'utilisation « valider un compte ».....	33

ABBREVIATIONS

WWW: Wide World Web.

HTML: Hyper Text markup language.

UML: Unified Modeling Languages.

UP: unified process.

IHM : Interface Homme-Machine.

JS : java script.

CSS: Cascading Style Sheets.

PHP: Personnel Home Page.

CMS: Content Management System.

HTTP: Hyper text Transfer Protocol.

SMTP: Simple mail transfer protocol

FTP: file transfer protocol

Introduction générale

1 Contexte du travail

Aujourd'hui, l'internet est présent dans tous les secteurs, cette omniprésence est un bouleversement en soit, plus important que celui de l'informatique en son temps, même s'il en est le prolongement logique. Le but de l'internet n'est plus comme à l'origine, de transmettre quelques lignes de message, mais d'échanger des documents électroniques, des données informatisées, des informations économiques, des schémas, des sons...etc. Ainsi, et en quelques années, la notoriété d'Internet est passée de la simple découverte à une explosion de services qui intéressent les professionnels comme les particuliers.

Dans le domaine économique, Internet se présente comme un outil d'information de tout premier plan, offrant aux entreprises de nombreux services interactifs : marketing en direct, publicités, affiche commercial ...etc. Ceci à l'aide des **Nouvelles Technologies d'Information est de Communication**.

Actuellement, les **Nouvelles Technologies d'Information est de Communication, NTIC**, ouvrent, largement, la porte d'information et de communication devant toutes les entreprises qui ne veulent pas rater l'occasion d'avoir une place dans un marché dont la clientèle est constituée du monde entier. Ainsi, les **NTIC** approchent les entreprises de leurs clients et leurs permettent de bien apporter les services chez eux.

2 Problématique et objectif

L'immobilier a une très grande importance dans l'économie de tout pays. Les technologies du web ont été beaucoup exploitées dans les pays développés afin de faciliter le travail des agents immobiliers, ce qui n'est pas le cas encore en Algérie sauf de manière assez rudimentaire. Les sites web algériens de ce domaine restent peu évolués et ils n'utilisent pas toutes capacités potentielles des nouvelles technologies du web.

Nous avons abordé dans le cadre de notre travail un sujet concernant la réalisation d'un site web pour une agence immobilière virtuelle **EI-DAR**. Notre système offre une panoplie de services que nous avons répertoriés suivant les espaces que nous offrons à nos principaux acteurs : *Espace Client* : dans cet espace, le client peut effectuer la recherche automatique des biens, consulter les informations de chacun des biens existants et faire des transactions ultérieurement. *Espace propriétaire* : dans cet espace, le propriétaire peut déposer des biens qu'il aimerait soumettre en vente ou en location après inscription dans le site. *Espace Administrateur* : dans cet espace, l'administrateur est chargé de faire la gestion du site, à savoir : les biens et comptes.

Pour la conception de notre application, nous avons suivi un processus adapté au développement des applications web. Pour représenter les différents aspects de notre application selon la conception spécifiée, nous avons utilisé le langage de

modélisation UML, qui est outil très puissant dans le domaine de développement des applications web.

Enfin, pour l'implémentation du site web, nous avons adopté le langage PHP, qui est l'un des langages de développement web les plus répandus pour la réalisation des pages web dynamiques.

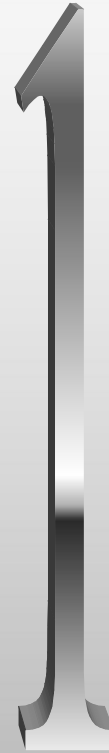
3 Organisation du mémoire

Ce mémoire est constitué de quatre chapitres :

- Dans le **premier chapitre**, nous présentons des généralités sur l'internet et les réseaux en général et sur le WWW en particulier.
- Le **deuxième chapitre** présente une description du langage de modélisation UML ainsi que le processus utilisé pour le développement de l'application.
- Le **troisième chapitre** nous allons décrire de façon détaillée les différentes phases que nous avons suivies pour parvenir à la réalisation de ce logiciel. Nous avons décrit en premier lieu l'étape d'identification et de spécifications des besoins, par la suite nous sommes passés à la phase de conception.
- Le **quatrième chapitre** présente les différents langages de programmation et les outils utilisés pour l'implémentation de notre application.

Ce mémoire s'achève par une conclusion générale qui spécifie ce qui a été effectivement réalisé.

CHAPITRE I : TECHNOLOGIE
WEB



1.1 Introduction

S'il est un phénomène qui a marqué le monde de l'informatique, c'est bien celui d'Internet. Ce réseau mondial créé par l'armée américaine, puis utilisé par les chercheurs et autres scientifiques, a connu une croissance phénoménale auprès du grand public avec l'introduction du Word Wide Web.

Cependant, il est nécessaire d'avoir un minimum de notion sur la technologie internet et le web, ce qui est l'objectif du présent chapitre.

1.2 Internet [1]

1.2.1 Qu'est ce qu'internet ?

Il existe plusieurs définitions dans la littérature. On cite quelques une :

- **Définition 1** : Internet est un réseau international d'ordinateurs qui communiquent entre eux grâce à des protocoles d'échanges de données standard. Cette communication en réseaux se fait indépendamment des types d'ordinateurs utilisés (Mac, PC, Unix ou autres). Internet est un outil de communication qui utilise les fils téléphoniques, les fibres optiques, les câbles intercontinentaux et les communications par satellite.

- **Définition 2** : Internet est un système d'interconnexion de machines qui constitue un réseau informatique mondial, utilisant un ensemble standardisé de protocoles de transfert de données.

C'est un réseau de réseaux, composé de millions de réseaux aussi bien publics que privés, universitaires, commerciaux et gouvernementaux. Internet transporte un large spectre d'information et permet l'élaboration d'applications et de services variés comme le courrier électronique, la messagerie instantanée et le World Wide Web.

Internet ayant été popularisé par l'apparition du World Wide Web, les deux sont parfois confondus par le public non averti. Le World Wide Web n'est pourtant que l'une des applications d'Internet.



Figure 1.1 : interconnexion entre les ordinateurs

Il faut savoir qu'Internet est un réseau composé d'ordinateurs. Ceux-ci peuvent être classés en deux catégories.

- **Les clients** : ce sont les ordinateurs des internautes. Chaque client représente un visiteur d'un site web.
- **Les serveurs** : ce sont des ordinateurs puissants qui stockent et délivrent des sites web aux internautes, c'est-à-dire aux clients. La plupart des internautes n'ont jamais vu un serveur de leur vie. Pourtant, les serveurs sont indispensables au bon fonctionnement du Web.

1.2.2 Histoire d'Internet

L'histoire d'Internet remonte au développement des premières télécommunications. L'idée d'un réseau informatique, permettant aux utilisateurs de différents ordinateurs de communiquer, se développa par de nombreuses étapes successives. La somme de tous ces développements conduisit au « réseau des réseaux » (*network of networks*) que nous connaissons aujourd'hui en tant qu'Internet. Il est le fruit à la fois de développements technologiques et du regroupement d'infrastructures réseau existantes et de systèmes de télécommunications.

Les premières versions mettant en place ces idées apparurent à la fin des années 1950. L'application pratique de ces concepts commença à la fin des années 1960. Dès les années 1980, les techniques que nous reconnaissons maintenant comme les fondements d'Internet moderne commencèrent à se répandre autour du globe. Dans les années 1990, sa popularisation passa par l'apparition du World Wide Web

L'infrastructure d'Internet se répandit autour du monde pour créer le large réseau mondial d'ordinateurs que nous connaissons aujourd'hui. Il se répandit au travers des pays occidentaux puis frappa à la porte des pays en voie de développement, créant ainsi un accès mondial à l'information et aux communications sans précédent ainsi qu'une fracture numérique. Internet contribua à modifier fondamentalement l'économie mondiale, y compris avec les retombées de la bulle Internet.

1.3 Le Web

1.3.1 World Wide Web (WWW) [1]

Le World Wide Web (WWW), littéralement la « toile (d'araignée) mondiale », communément appelé le Web, et parfois la Toile, est un système hypertexte public fonctionnant sur Internet.

Le Web permet de consulter, avec un navigateur, des pages accessibles sur des sites. L'image de la toile d'araignée vient des hyperliens qui lient les pages web entre elles.



Figure 1.2 : Logo historique du World Wide Web par Robert Cailliau.

1.3.2 Site Web

1.3.2.1 Quelques définitions

- **Un site Web** est un regroupement de pages web1 sur un sujet, un thème, un commerce ou une organisation. Un site Web a aussi une page principale. C'est une page Web qui aide les lecteurs à naviguer sur le site pour trouver l'information voulue.

Un site Web doit aussi être structuré. Comment une page Web est-elle reliée à une autre? Y a-t-il un ou plusieurs chemins ou parcours que les lecteurs peuvent utiliser pour naviguer à travers le site? Par exemple. Au début et à la fin de chaque page, il y a plusieurs boutons de navigation pour passer à la page précédente ou suivante.

- **Le client Web (navigateur) [1]**

On appelle logiciel client un programme qui utilise le service offert par un serveur. Le client envoie une requête et reçoit la réponse. Un navigateur Web est un logiciel client HTTP conçu pour accéder aux ressources du Web.

Sa fonction de base est de permettre la consultation des documents HTML disponibles sur les serveurs HTTP. Le support de d'autres types de ressource et d'autres protocoles de communication dépend du navigateur considéré. Exemples de navigateurs internet : Firefox, Google chrome, Internet Explorer de Microsoft.

- **Le serveur web [1]**

Un serveur web est un ordinateur connecté à Internet et sur lequel sont hébergés des sites web, composés de pages HTML (le serveur web, également appelé serveur HTTP, peut également Une **page Web** est un fichier qui contient du texte, des images et des liens à d'autres pages.

être composé d'un groupe d'ordinateurs). Le logiciel fédérateur, sur un serveur web, est le serveur HTTP (Apache, le plus fréquemment), auquel viennent s'adjoindre un

interpréteur de langage dynamique (PHP dans la plupart des cas), un gestionnaire de base de données (tel que

MySQL) et d'autres programmes, comme un serveur SMTP.

La fonction d'un serveur web est de répondre aux requêtes des navigateurs Internet (Internet Explorer, Firefox , Chrome , Opéra, Safari...)

- **La requête** : C'est un message envoyé par le client au serveur décrivant l'opération à exécuter.
- **La réponse** : C'est le message envoyé par le serveur à un client (l'exécution d'une requête).

1.3.2.2 Les types de site web [2]

Il existe deux types de site web : les sites statiques et les sites dynamiques.

1.3.2.2.1 Les sites web statiques

a. Définition

Ce sont des sites réalisés uniquement à l'aide des langages (X) HTML et CSS. Ils fonctionnent très bien mais leur contenu ne peut pas être mis à jour automatiquement: il faut que le propriétaire du site (le webmaster) modifie le code source pour y ajouter des nouveautés. Ce n'est pas très pratique quand on doit mettre à jour son site plusieurs fois dans la même journée! Les sites statiques sont donc bien adaptés pour réaliser des sites "vitrine", pour présenter par exemple son entreprise, mais sans aller plus loin.

Ce type de site se fait de plus en plus rare aujourd'hui, car dès que l'on rajoute un élément d'interaction (comme un formulaire de contact), on ne parle plus de site statique mais de site dynamique. Lorsque le site est statique, le schéma est très simple. Cela se passe en deux temps :

- Le client demande au serveur une page Web.
- Le serveur lui répond en lui envoyant la page réclamée.



Figure 1.3 : site statique

b. Quels avantages pour un site statique ?

On vient de voir qu'un site statique possède beaucoup d'inconvénient : il faut s'y connaître en HTML pour le modifier et l'étape de mise à jour est fastidieuse. (Pour résumer) Mais il faut aussi reconnaître au site statique des avantages dans plusieurs domaines :

- le site internet est mis à jour en local sur la machine de l'administrateur : il n'y a donc pas de surprise une fois que le site est en ligne.
- le site internet ne fait pas appel aux technologies en perpétuelles évolutions qui permettent la mise en place de sites dynamique (PHP, Ruby, Python, Perl, Java, ASP, etc.) : on gagne donc en sécurité et en veille technologique.
- le site internet statique consomme peu de ressource serveur : le site n'utilisant aucune technologie compliquée (au hasard : PHP + MySQL + Apache), les coûts d'entretien et de maintenance en activité sont très inférieurs à ceux d'un site dynamique.
- le site internet statique se sauvegarde plus facilement : ceux qui ont déjà manipulé les bases de données MySQL utilisées pour la création de sites dynamiques savent que c'est une galère à sauvegarder et à restaurer. Le fait de disposer directement des pages HTML du site facilite la sauvegarde (un simple copier / coller sur une clé USB est c'est bon !).

1.3.2.2 Les sites web dynamiques

a. Définition

Plus complexes, ils utilisent d'autres langages en plus de HTML et CSS, tels que PHP et MySQL. Le contenu de ces sites web est dit « dynamique » parce qu'il peut changer sans l'intervention du webmaster ! La plupart des sites web que vous visitez aujourd'hui, y compris le Site du Zéro, sont des sites dynamiques. Le seul pré requis pour apprendre à créer ce type de sites est de déjà savoir réaliser des sites statiques en HTML et CSS.



Figure 1.4 : site dynamique

b. Quels avantages pour un site dynamique ?

- De nombreux scripts gratuits existent déjà et permettent de réaliser tous les sites qu'on souhaite. Ainsi en téléchargeant le script (ou CMS : Content Management System) qui va bien, il sera très simple de créer un forum, un blog ou tout autre site.
- La mise à jour est très simple : une fois le script dynamique en place, on met à jour le site en ligne dans la partie « administration » du site. On peut donc mettre à jour le site de n'importe quel ordinateur et même depuis certains téléphones mobiles (avec accès Internet naturellement).
- Avec un site dynamique il est possible de réaliser une grande interaction avec les visiteurs : les visiteurs peuvent donc rester beaucoup plus longtemps sur vos pages si les fonctionnalités sont intéressantes.

1.3.2.2.3 Quel type de site pour quelle utilisation ?

On utilisera un site web statique pour une utilisation bien particulière. On utilisera ce fonctionnement pour un site web nécessitant peu de maintenance, peu de mise à jour et contenant peu de pages. En effet, comme la mise à jour d'un site Internet statique peut être fastidieuse, on utilisera ce type de site uniquement si les mises à jour sont exceptionnelles. Car, à chaque mise à jour il faudra modifier la page HTML du site et la mettre en ligne en effectuant une copie par FTP.

On utilisera plutôt un site dynamique si on souhaite créer une interaction avec ses lecteurs. Le site dynamique permet de se connecter en ligne sur son site pour réaliser sa mise à jour en direct. Une fois la mise à jour du site dynamique effectuée, le résultat apparaît directement aux lecteurs. On privilégiera donc les sites web dynamiques pour les sites permettant aux

visiteurs de laisser des commentaires (blogs) ou de converser avec d'autres lecteurs (forums). De même si le site doit être mis à jour très fréquemment (plusieurs fois par semaine) on pourra choisir de créer un site dynamique même si aucune interaction n'est prévue avec les visiteurs.

Par exemple pour réaliser un site vitrine présentant les 10 produits vendus par une entreprise on pourra choisir :

- **Un site statique** si les produits ne sont modifiés qu'une ou deux fois par mois et qu'un ou deux nouveaux produits sont ajoutés au catalogue chaque année.
- **Un site dynamique** si on souhaite que les visiteurs ajoutent des commentaires sur les fiches produits et/ou qu'il faut modifier très souvent (ajout, suppression, modification) les fiches produits.

1.4 Conclusion

Nous détaillons dans le prochain chapitre les concepts de base du langage de modélisation UML, avec la démarche de modélisation utilisée.

**CHAPITRE 2 : LANGAGE DE
MODELISATION UNIFIE
UML**

2.1 Introduction

Ce chapitre présente en premier lieu les concepts de base du langage de modélisation UML (Unified Modeling Language). Nous commençons par une description d'UML, en suite, nous citons quelques caractéristiques qui le rendent le langage de modélisation le plus répandu.

Ainsi, nous énumérons ses diagrammes avec une description de ceux les plus utilisés dans la modélisation. Cette étude permette une bonne lecture de ce qui va suivre et exprime de manière uniforme l'analyse, la conception et la réalisation d'une application informatique.

Finalement, nous allons présenter une démarche de modélisation simplifiée et adaptée au développement des sites web.

2.2 Présentation du langage de modélisation UML

2.2.1 Historique d'UML [3]

Au début des années 90, une cinquantaine de méthodes objet ont vu le jour. Ce qui est un signe de l'intérêt du sujet, et également de confusion. Toutes ces méthodes utilisaient à peu près les mêmes concepts de classes d'association, de partition en sous-systèmes.

En octobre 1994, Grady Booch et Jim Rumbaugh fondèrent la Rational Software corporation Pour unir leurs efforts en vue de créer une norme industrielle unique à partir de leurs Méthodes. C'est ainsi qu'est née la méthode unifiée (Unified Method) en octobre 1995. Ils sont ensuite rejoints par Evar Jacobson. L'inventeur des cas d'utilisation (Use cases), et ont publié UML 0.9 en juin 1996, puis UML 1.0 en janvier 1997.

La version 1.1 mise au point avec d'autres partenaires, devient un standard en novembre 1997 lorsqu'est acceptée par l'OMG (Object Management Group). La poursuite du Développement d'UML fut ensuite intégralement remise aux mains de l'OMG. En juillet 1998, L'OMG publiait UML 1.2, puis UML 1.3 en juin 1999. En mai 2002 UML 1.4, qui présentait de petites améliorations et quelque extension, voyait le jour UML 1.5 a été publié en mars 2003 avec également quelques correctifs, Version 2.0 en septembre 2004, Version 2.4.1 en août 2011.

2.2.2 Définition d'UML

Uml (Unified Modeling language) traduit « langage de modélisation objet unifié », il s'agit d'un langage graphique de modélisation objet permettant de spécifier, de construire et de visualiser les détails d'un système logiciel. L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.

Historique d'UML

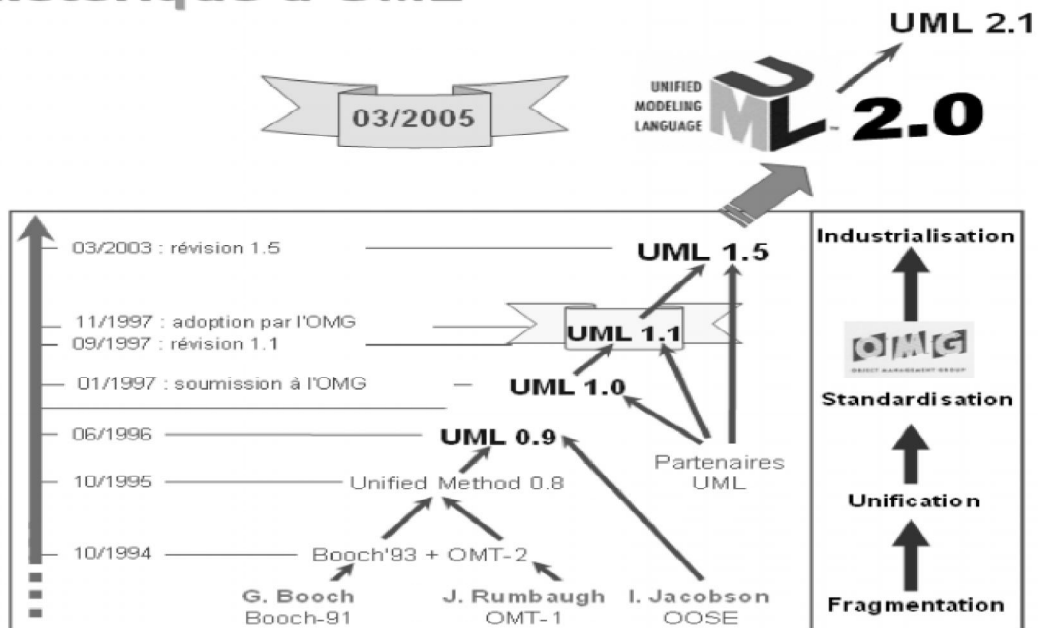


Figure 2.1 : historique d'UML

2.2.3 Pourquoi UML ? [4]

De la même façon qu'il vaut mieux dessiner une maison avant de la construire, il vaut mieux modéliser un système avant de le réaliser.

UML pour :

- Obtenir une modélisation de très haut niveau indépendamment de tout langage De programmation.
- Faire collaborer des participants de tous horizons autour d'un même document de synthèse.
- Faire des simulations avant de construire un système.
- Exprimer dans un seul modèle tous les aspects statiques, dynamiques, juridiques, Spécifications, etc.
- Documenter un projet.
- Générer automatiquement la partie logicielle d'un système.

2.2.4 Présentation des vues et diagrammes d'UML [5]

Le méta-modèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relie. Toutefois, Étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux **vues**. Une vue est constituée d'un ou plusieurs diagrammes.

2.2.4.1 la vue statique

Une vue d'un système qui met l'accent sur la structure des objets, avec leur classificateurs, leurs relations, leurs attributs et leurs opérations. Cette vue comporte les diagrammes suivants :

1. Diagramme de classe
2. Diagramme d'objet
3. Diagramme de composants
4. Diagramme de déploiement
5. Diagramme de paquetage
6. Diagramme de structure composite

2.2.4.2 la vue dynamique

Met l'accent sur le comportement dynamique du système en montrant la collaboration entre les objets et les modifications apportées à l'état interne des objets, cette vue inclut les diagrammes suivants :

1. Diagramme d'activité ;
2. Diagramme des cas d'utilisation ;
3. Diagramme de communication ;
4. Diagramme d'état-transition ;
5. Diagramme globale d'interaction ;
6. Diagramme de séquence ;
7. Diagramme de temps

2.2.5 Description de quelques diagrammes UML

2.2.5.1 Le diagramme de cas d'utilisation

1 Définition : Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. Le diagramme de cas d'utilisation décrit la succession des opérations réalisées par un acteur (personne qui assure l'exécution d'une activité). C'est le diagramme principal du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en oeuvre.

2 Éléments constitutifs des cas d'utilisation

a- Acteur : entité externe qui agit sur le système. Le terme acteur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes. les acteurs sont des classificateurs qui représentent des rôles au travers d'une certaine utilisation (cas) et non pas des personnes physiques. Ce sont des acteurs types.

b- Cas d'utilisation : ensemble d'actions réalisées par le système en réponse à une action d'un acteur.

c- Relations dans les diagrammes de cas d'utilisation : Après avoir identifié les acteurs et les cas d'utilisation, il est utile de restructurer l'ensemble des cas

d'utilisation que l'on a fait apparaître afin de rechercher les comportements partagés, les cas particuliers et les généralisations.

- **Les relations entre les acteurs** : La seule relation entre les acteurs est la relation de généralisation.
- **Les relations possibles entre cas d'utilisation** : UML définit trois types de relations standardisées entre cas d'utilisation, détaillées ci-après :
 - **La relation d'inclusion** : Lors de la description des cas d'utilisation, il apparaît qu'il existe des sous-ensembles communs à plusieurs cas d'utilisation, il convient donc de factoriser ces fonctionnalités en créant de nouveaux cas d'utilisation qui seront utilisés par les cas d'utilisation qui les avaient en commun. Un cas d'utilisation A utilise un cas d'utilisation B signifie que une instance de A va engendrer une instance de B et l'exécuter, A dépend de B, B n'existe pas tout seul et A n'existe pas sans B.
 - **La relation d'extension** : Les extensions (Extend) représentent des prolongements logiques de certaines tâches sous certaines conditions. Autrement dit un cas d'utilisation A étend un cas d'utilisation B lorsque le cas d'utilisation A peut être appelé au cours de l'exécution du cas d'utilisation B. Elle est représentée par une flèche en pointillée avec le terme Extend. Ce type de relation peut être utile pour traiter des cas particuliers ou préciser les objectifs, ou pour tenir compte de nouvelles exigences au cours de la maintenance du système et de son évolution.
 - **Relation de généralisation entre cas d'utilisation** : La troisième relation est la relation de généralisation ou spécialisation. Le cas d'utilisation A est une généralisation de B, si B est un cas particulier de A c'est-à-dire lorsque A peut être substitué par B pour un cas précis. Ces relations sont des traits pleins terminés par une flèche en triangle.

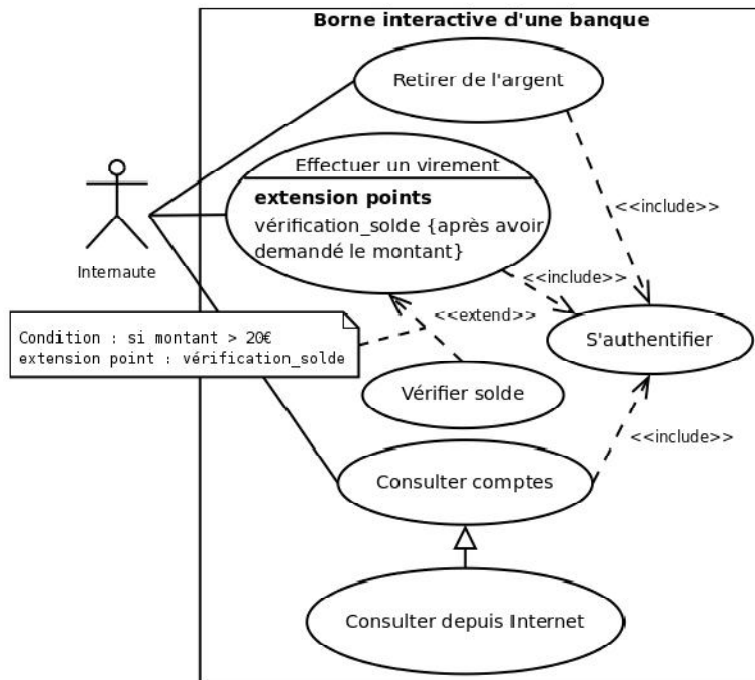


Figure 2.2 : diagramme de cas d'utilisation

d- La description textuelle des cas d'utilisation [5] : Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. Bien que de nombreux diagrammes d'UML permettent de décrire un cas, il est recommandé de rédiger une description textuelle car c'est une forme souple qui convient dans bien des situations.

2.2.5.2 Le diagramme de classe

1 définition : Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. Une classe permet de décrire un ensemble d'objets (attributs et comportement), tandis qu'une relation ou association permet de faire apparaître des liens entre ces objets. On peut donc dire :

- un objet est une instance de classe.
- un lien est une instance de relation.

Le diagramme de classe est un modèle permettant de décrire de manière abstraite et générale les liens entre objets. Il comporte : les classes, les attributs, les identifiants, les relations les opérations ...etc.

UML définit trois niveaux de visibilité pour les attributs et les opérations :

1-Public qui rend l'élément visible à tous les clients de la classe.

2-Protégé qui rend l'élément visible aux sous classes de la classe.

3-Privé qui rend l'élément visible à la classe seule

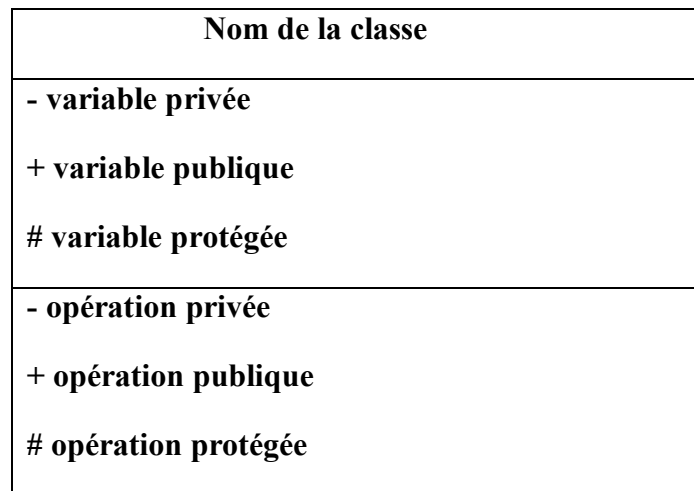


Figure 2.3 : représentation graphique d'une classe

2- Éléments constitutifs de diagramme de classe

a- Nom d'une classe : Le nom de la classe doit évoquer le concept décrit par la classe. Il commence par une majuscule. On peut ajouter des informations subsidiaires comme le nom de l'auteur de la modélisation, la date, etc. Pour indiquer qu'une classe est abstraite, il faut ajouter le mot-clef `abstract`.

b- Attributs de la classe : Les attributs définissent des informations qu'une classe ou un objet doivent connaître. Ils représentent les données encapsulées dans les objets de cette classe.

Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisé. Le nom de l'attribut doit être unique dans la classe.

c- Méthode de la classe : Dans une classe, une opération (même nom et même types de paramètres) doit être unique. Quand le nom d'une opération apparaît plusieurs fois avec des paramètres différents, on dit que l'opération est surchargée. En revanche, il est impossible que deux opérations ne se distinguent que par leur valeur retournée

d- Relations entre classes : sont les suivantes :

- **Les associations :** Une association représente une relation structurelle entre classes d'objets. La plupart des associations sont binaires, c'est à dire qu'elles connectent deux classes. On représente une association en traçant une ligne entre les classes associées.
- **Les classes associations :** Il peut arriver que l'on ait besoin de garder des informations (attributs ou opérations) Propres à une association. Une classe de

ce type est appelée classe association. Une classe association est une classe comme une autre qui peut entretenir des relations avec d'autres classes.

- **Agrégation** : Une agrégation est un type particulier d'association. Elle traduit la volonté de renforcer la dépendance entre les classes. C'est une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité. Les critères suivants impliquent une agrégation :
 - une classe fait partie d'une autre classe.
 - une action sur une classe implique une action sur une autre classe.
 - les objets d'une classe sont subordonnés aux objets d'une autre classe.
- **La composition** : La composition est un cas particulier d'agrégation dans laquelle la vie des composants est liée à celle de l'agrégat. Dans la composition, l'agrégat ne peut être multiple. La composition se représente par un losange noir. Une composition est une association contraignante : la suppression d'un objet agrégat entraîne la suppression des objets agrégés.
- **Généralisation** : UML emploie le terme de généralisation pour désigner la relation de classification entre un élément plus général et un élément plus spécifique. La relation de généralisation signifie «est un » ou « est une sorte de»

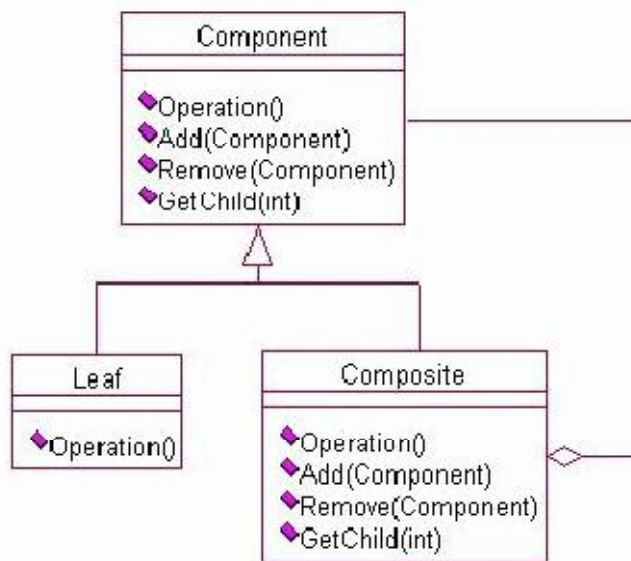


Figure 2.4 : diagramme de classe

2.2.5.3 Diagramme de séquence

1 Définition : Le diagramme de séquences permet de cacher les interactions d'objets dans le cadre d'un scénario d'un diagramme de cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

2 Éléments constitutifs de diagramme de séquence

a- Ligne de vie : Une ligne de vie représente l'ensemble des opérations exécutées par un objet. Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information peut être implicite (cas général) ou explicite à l'aide d'un message de retour.

b- Message synchrone et asynchrone : Dans un diagramme de séquence, deux types de messages peuvent être distingués :

- **Message synchrones :** l'émetteur reste bloqué le temps que le récepteur traite le message envoyé et envoie la réponse. Graphiquement, un message synchrone se représente par une flèche en traits pleins et à l'extrémité pleine, suivi d'une réponse qui se représente par une flèche en discontinue.
- **Message asynchrones :** L'émetteur n'est pas bloqué lorsque le récepteur traite le message envoyé. Graphiquement, un message asynchrone se représente par une flèche en traits pleins et à l'extrémité ouverte.

c- Message de création et destruction d'instance :

- La création d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie.
- La destruction d'un objet est matérialisée par une croix qui marque la fin de la ligne de vie de l'objet. La destruction d'un objet n'est pas nécessairement consécutive à la réception d'un message.

d- Les opérateurs : Les principaux opérateurs de diagramme de séquences sont:

- **Loop (boucle):** Le fragment peut s'exécuter plusieurs fois, et la condition de garde explicite l'itération.
- **Opt (optionnel) :** Le fragment ne s'exécute que si la condition fournie est vraie.
- **Alt (fragments alternatifs):** Seul le fragment possédant la condition vraie s'exécutera.

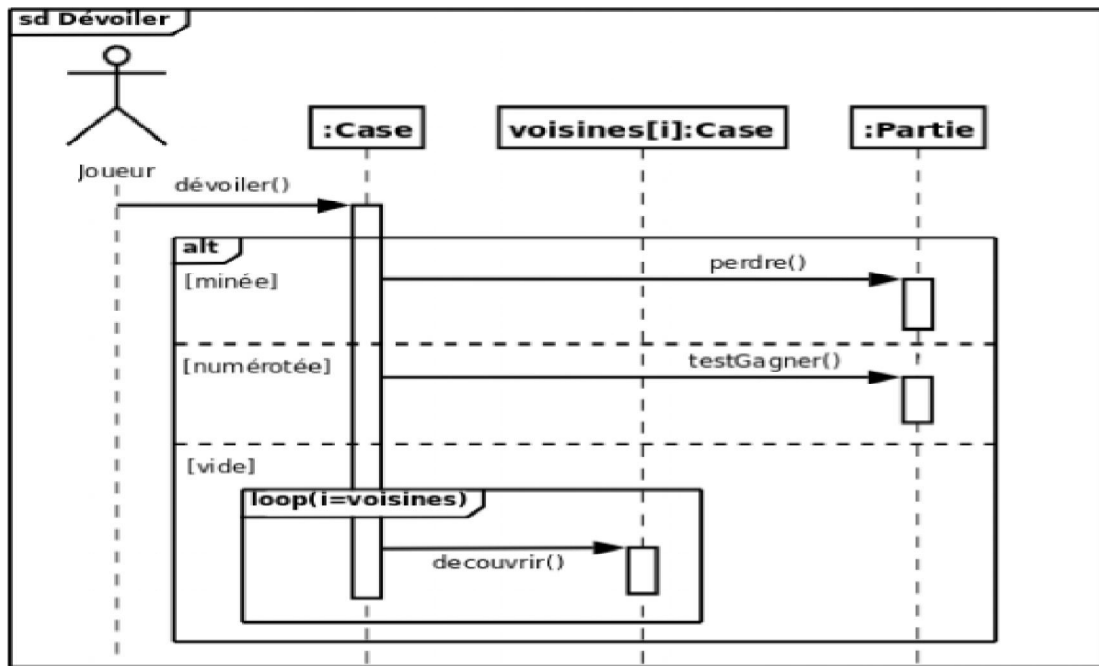


Figure 2.5 : de diagramme de séquence

2.2.5.4 Diagramme d'activités

1 Définition : Le diagramme d'activité n'est autre que la transcription dans UML de la représentation du processus telle qu'elle a été élaborée lors du travail qui a préparé la modélisation. Il montre l'enchaînement des activités qui concourent au processus.

2 les éléments constitutifs

a- Activité : Une activité UML représentée par un rectangle aux coins arrondis et contient la description textuelle des actions de base qu'elle réalise, ou simplement son nom si le niveau de spécification n'est pas encore assez précis pour détailler les actions, aucun syntaxe spécifique n'est proposée dans la norme pour l'expression de ces actions : on utilise le plus souvent une syntaxe proche d'un langage de programmation ou à défaut du pseudo-code.

b- Transition : les activités liées par des transitions représentées par des arcs orientés pouvant porter des gardes qui représentent le cheminement du flot de control de l'application.

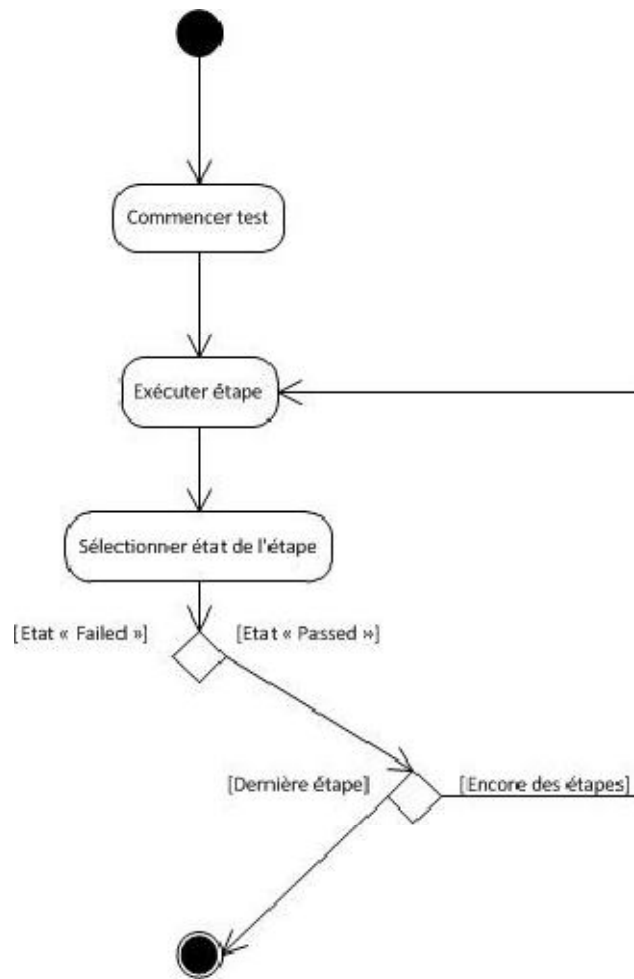


Figure 2.6 : Représentation graphique d'un diagramme d'activité

2.2.6 Avantages et inconvénients d'UML

2.2.6.1 Les points forts d'UML [6]

UML est un langage formel et normalisé :

- gain de précision.
- gage de stabilité.
- encourage l'utilisation d'outils.

UML est un support de communication performant :

- Il cadre l'analyse.
- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

2.2.6.2 Les points faibles d'UML [6]

- Les diagrammes UML doivent être synchronisés avec le code du logiciel, qui nécessite du temps pour mettre en place et à entretenir, et ajoute travail à un projet de développement logiciel.

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation, la nécessité de s'accorder sur des modes d'expression communs est vitale en informatique.
- UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue. Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard.

2.3 Démarche de modélisation [7]

UML n'est pas une méthode et ne propose pas une démarche de modélisation explicitant et encadrant toutes les étapes d'un projet. Il n'est qu'un langage de modélisation ; ce qui justifie le besoin de faire appel à une méthode qui permet le passage de l'expression des besoins au code de l'application, nous allons donc présenter une des méthodes proposées dans la littérature. Il s'agit d'une méthode simple et générique proposée par LAURENT AUDIBERT. Elle se situe à mi-chemin entre UP (unified process), qui constitue un cadre général très complet de processus de développement et XP qui est une approche minimaliste à la mode centrée sur le code.

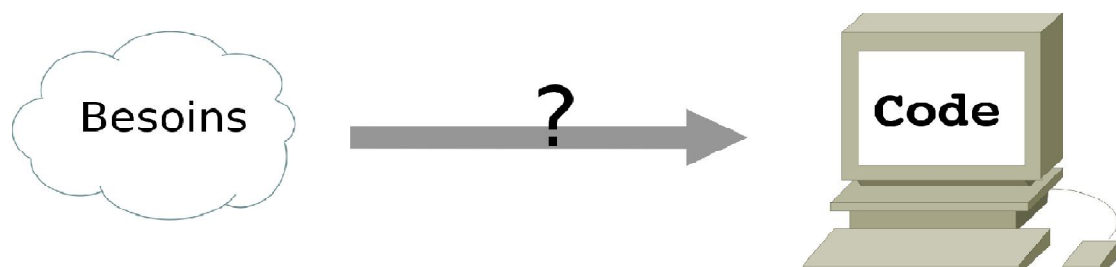


Figure 2.7 : Quelle méthode pour passer de l'expression des besoins au code de l'application ?

2.3.1 Identification des besoins et spécification des fonctionnalités

2.3.1.1 Identification et représentation des besoins : Diagramme de cas d'utilisation

Les cas d'utilisation sont utilisés tout au long du projet. Dans un premier temps, on les crée pour identifier et modéliser les besoins des utilisateurs. Ces besoins sont déterminés à partir des informations recueillies lors des rencontres entre informaticiens et utilisateurs. Il faut impérativement proscrire toute considération de réalisation lors de cette étape.

Durant cette étape, il faut déterminer les limites du système, identifier les acteurs et recenser les cas d'utilisation. Si l'application est complexe, vous pourrez organiser les cas d'utilisation en paquetages.

2.3.1.2 Spécification détaillée des besoins : diagrammes de séquence système

Dans cette étape, on cherche à détailler la description des besoins par la description textuelle des cas d'utilisation et la production de diagrammes de séquence système illustrant cette description textuelle. Cette étape amène souvent à mettre à jour le diagramme de cas d'utilisation puisque nous sommes toujours dans la spécification des besoins.

Les scénarios de la description textuelle des cas d'utilisation peuvent être vus comme des instances de cas d'utilisation et sont illustrés par des diagrammes de séquence système. Il faut au minimum, représenter le scénario nominal de chacun des cas d'utilisation par un diagramme de séquence qui rend compte de l'interaction entre l'acteur, ou les acteurs, et le système.

2.3.1.3 Maquette de l'IHM de l'application (non couvert par UML)

Une maquette d'IHM (Interface Homme-Machine) est un produit jetable permettant aux utilisateurs d'avoir une vue concrète mais non définitive de la future interface de l'application. La maquette peut très bien consister en un ensemble de dessins produits par un logiciel de présentation ou de dessin. Par la suite, la maquette pourra intégrer des fonctionnalités de navigation permettant à l'utilisateur de tester l'enchaînement des écrans ou des menus, même si les fonctionnalités restent fictives. La maquette doit être développée rapidement afin de provoquer des retours de la part des utilisateurs.

2.3.2 Phases d'analyse

2.3.2.1 Analyse du domaine (modèle du domaine)

La modélisation des besoins par des cas d'utilisation s'apparente à une analyse fonctionnelle classique. L'élaboration du modèle des classes du domaine permet d'opérer une transition vers une véritable modélisation objet. L'analyse du domaine est une étape totalement dissociée de l'analyse des besoins. Elle peut être menée avant, en parallèle ou après cette dernière. La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes appelée modèle du domaine.

Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine (on utilise aussi le terme de métier) de l'application.

2.3.2.2 Diagramme de classes participantes

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la

Jonction entre, d'une part, les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de conception logicielle que sont les diagrammes d'interaction et le diagramme de classes de conception. Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation.

- **Les classes de dialogues** : Les classes qui permettent les interactions entre l'IHM et les utilisateurs sont qualifiées de dialogues. Ces classes sont directement issues de l'analyse de la maquette. Il y a au moins un dialogue pour chaque association entre un acteur et un cas d'utilisation du diagramme de cas d'utilisation. En général, les dialogues vivent seulement le temps du déroulement du cas d'utilisation concerné.
- **Les classes de contrôles** : Les classes qui modélisent la cinématique de l'application sont appelées contrôles. Elles font la jonction entre les dialogues et les classes métier en permettant aux différentes vues de l'application de manipuler des informations détenues par un ou plusieurs objets métier, Elles contiennent les règles applicatives et les isolent à la fois des dialogues et des entités.
- **Les classes entités** : Les classes métier, qui proviennent directement du modèle du domaine, sont qualifiées d'entités. Ces classes sont généralement persistantes, c'est-à-dire qu'elles survivent à l'exécution d'un cas d'utilisation particulier et qu'elles permettent à des données et des relations d'être stockées dans des fichiers ou des bases de données. Lors de l'implémentation, ces classes peuvent ne pas se concrétiser par des classes mais par des relations, au sens des bases de données relationnelles.

2.3.2.3 Diagramme d'activités de navigation

Les IHM modernes facilitent la communication entre l'application et l'utilisateur en offrant toute une gamme de moyens d'action et de visualisation comme des menus déroulants ou contextuels, des palettes d'outils, des boîtes de dialogues, des fenêtres de visualisation, etc.

Cette combinaison possible d'options d'affichage, d'interaction et de navigation aboutis aujourd'hui à des interfaces de plus en plus riches et puissantes.

UML offre la possibilité de représenter graphiquement cette activité de navigation dans l'interface en produisant des diagrammes dynamiques. On appelle ces diagrammes des diagrammes de navigation. Les diagrammes d'activités de navigation sont à relier aux classes de dialogue du diagramme de classes participantes. Les différentes activités du diagramme de navigation peuvent être stéréotypées en fonction de leur nature : « fenêtre », « menu », « menu contextuel », « dialogue », etc. La modélisation de la navigation a intérêt à être structurée par acteur.

2.3.3 Phase de conception

2.3.3.1 Diagrammes d'interaction

Maintenant, il faut attribuer précisément les responsabilités de comportement, dégagée par Le diagramme de séquence système, aux classes d'analyse du diagramme de classes participantes. Les résultats de cette réflexion sont présentés sous la forme de diagrammes d'interaction UML Inversement, l'élaboration de ces diagrammes facilite grandement la réflexion. Parallèlement, une première ébauche de la vue statique de conception, c'est-à-dire du diagramme de classes de conception, est construite et complétée.

Durant cette phase, l'ébauche du diagramme de classes de conception reste indépendante des choix technologiques qui seront faits ultérieurement.

Les diagrammes d'interactions sont particulièrement utiles au concepteur pour représenter graphiquement ces décisions d'allocations des responsabilités.

2.3.3.2 Diagramme de classes de conception

L'objectif de cette étape est de produire le diagramme de classes qui servira pour l'implémentation. Une première ébauche du diagramme de classes de conception déjà été élaborée en parallèle du diagramme d'interaction. Il faut maintenant le compléter en précisant les opérations privées des différentes classes. Il faut prendre en compte les choix techniques, comme le choix du langage de programmation, le choix des différentes bibliothèques utilisées (notamment pour l'implémentation de l'interface graphique) etc.

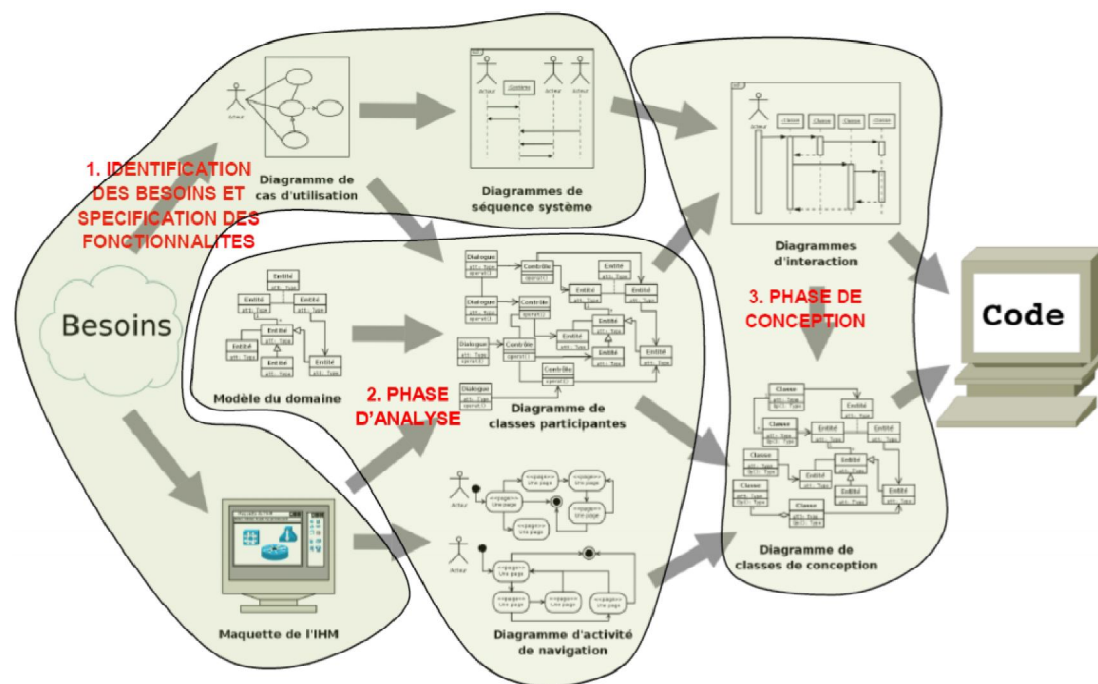


Figure 2.8 : Chaîne complète de la démarche de modélisation du besoin jusqu'au code

2.4 Conclusion

Ce chapitre nous permis d'étendre nos connaissances sur UML. Nous avons constaté qu'il est un langage riche, il couvre toutes les phases d'un cycle de développement. Il est également ouvert (indépendant du domaine d'application et des Langages d'implémentation), C'est pourquoi nous le choisis comme un langage de modélisation dans notre travail.

Dans le prochain chapitre nous passons à l'étude conceptuelle de notre site.

**CHPITRE 03 : ETUDE
DE CAS**



3.1. Introduction

Ce chapitre comporte la description des phases de la méthode présentée précédemment. Nous commençons par l'identification des besoins du système actuel qui permette de déterminer les différents acteurs ainsi que leurs tâches attribuées. Par la suite nous passons à l'étape d'analyse pour extraire le modèle du domaine, le diagramme des classes participantes et le diagramme d'activité de navigation.

3.2 Identification des besoins

Dans cette phase, nous allons réaliser les diagrammes suivants :

- Diagramme de cas d'utilisation ;
- Diagramme de séquence système ;

3.2.1 Diagramme de cas d'utilisation

3.2.1.1 Identification des acteurs

Dans notre projet il existe trois acteurs qui sont:

- **L'administrateur** : c'est la personne qui est responsable de la gestion des comptes des propriétaires et de faire la mise à jour du site.
- **Le propriétaire** : c'est la personne proposer et gère les annonces dans le site.
- **Le client** : c'est la personne qui est consulte et utiliser les annonce de site.

3.2.1.2 Diagramme de cas d'utilisation

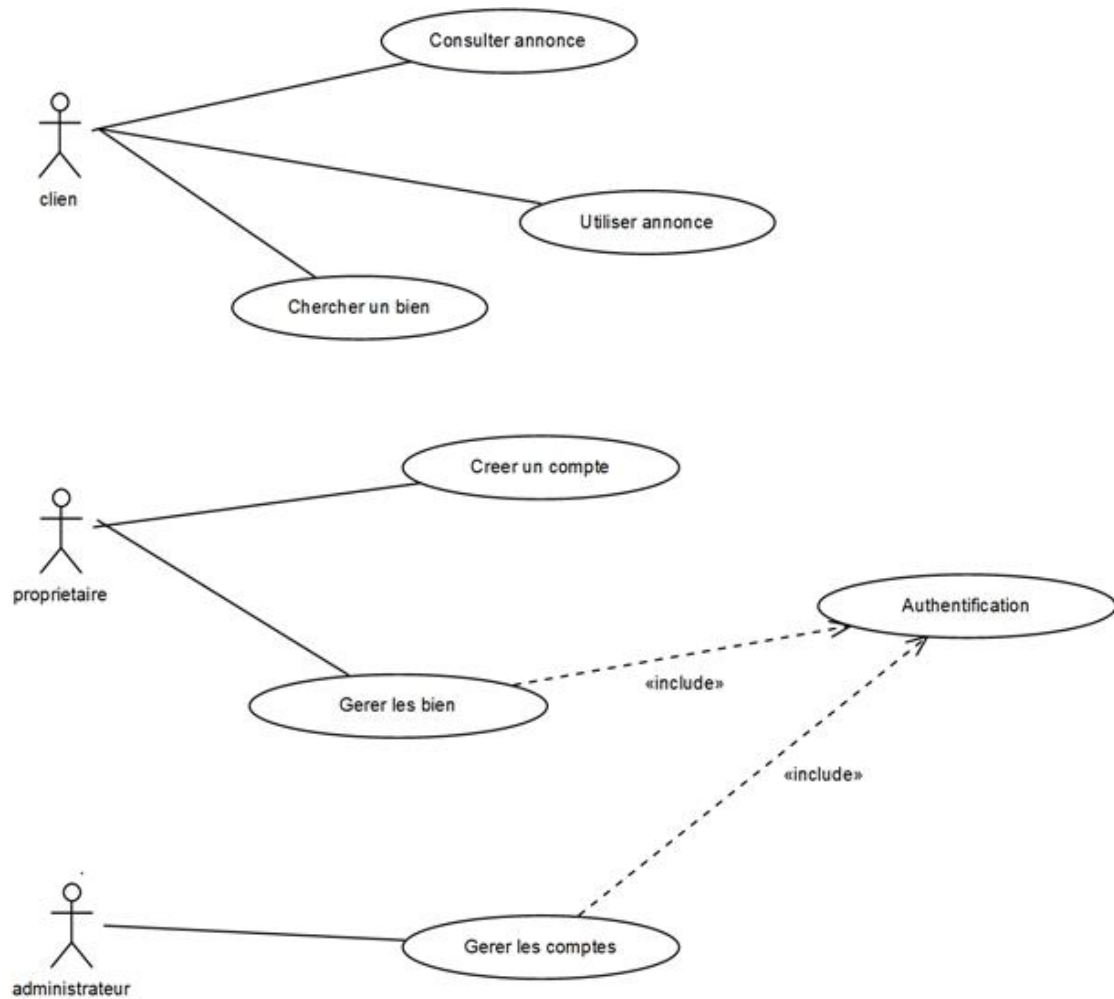


Figure 3.1 : diagramme de cas d'utilisation

3.2.1.3 Description textuelle des cas d'utilisation

3.2.1.3.1 Description du Cas d'utilisation « authentification »

Cas d'utilisation	Authentification.
Acteur	Utilisateur.
But	Le système identifie l'utilisation.
Pre_ condition	Néant.
Post_ condition	Le compte existe dans le site.
Scénario nominal	<p>1_ l' Utilisateur demande l'accès au site.</p> <p>2_ le système lui affiche un formulaire pour saisir le login et le mot de passe.</p> <p>3_ l'Utilisateur saisit le mot de passe.</p> <p>4_ le système effectue une vérification puis lance la page principale du site.</p>
Scénario alternatif	4_1.le système indique ou l'utilisateur que le compte n'existe pas et demande à l'utilisateur d'entrer le login et le

	mot de passe pour un deuxième fois. 4_2.reprise de l'enchaînement à partir l'étape 3.
--	--

Tableaux 3.1 : Description du Cas d'utilisation « authentification »

3.2.1.3.2 Description du cas d'utilisation « créer compte »

Cas d'utilisation	Créer un compte
Acteur	Propriétaire
But	Création d'un compte pour s'inscrire dans le site et déposer des annonces
Pre_ condition	Néant
Post_ condition	Le compte est crée par le système
Scénario nominal	1_ le Propriétaire demande au système la création d'un compte. 2_ le système lui envoie le formulaire d'inscription. 3_ le Propriétaire remplit le formulaire. 4_ le système confirme la création du compte.
Scénario alternatif	3_1.le formulaire rempli est incomplet ou incorrect. 3_2.le système indique au Propriétaire que le formulaire est incorrect, et demande au Propriétaire d'entrer une nouvelle fois les informations. 3_3.le compte existe déjà. 3_4.le système envoie un message d'erreur « compte déjà existe ». 3_5.reprise de l'enchaînement t à partir de l'étape 3.

Tableaux 3.2 : Description du cas d'utilisation « créer compte »

3.2.1.3.3 Description du Cas d'utilisation « Gérer un bien »

3.2.1.3.3 .1 « proposer un bien »

Cas d'utilisation	proposer un bien.
Acteur	Propriétaire
But	Pour poser un bien en vente ou en location.
Pre_ condition	Le propriétaire s'authentifier par le system.
Post_ condition	Configuration d'ajoute d'un bien.
Scénario nominal	1_ le système affiche la liste des options. 2_ le propriétaire sélectionne l'option « proposer un bien ». 3_ le système affiche un formulaire à remplir. 4_ le propriétaire remplit le formulaire. 5_ le système confirme l'ajout du bien.

Scénario alternatif	4_1.le formulaire rempli est incorrect. 4_2.le système indique au propriétaire que le formulaire est incorrect et demande au propriétaire d'entrer une nouvelle fois les informations. 4_3.reprise de l'enchaînement à partir du point 5.
---------------------	---

Tableaux 3.3 : Description du Cas d'utilisation « proposer un bien »

3.2.1.3.3 .2 « modifier un bien »

Cas d'utilisation	modifier un bien.
Acteur	Propriétaire.
But	Pou modifier les informations des biens.
Pre_ condition	le propriétaire s'authentifier par le système.
Post_ condition	Le bien immobilier est modifié.
Scénario nominal	1_ le système renvoie la liste des options. 2_ le propriétaire choisit l'option de modification. 3_ le système affiche la liste des biens. 4_ le propriétaire sélectionne le bien à modifier. 5_ le système affiche les informations qu'il peut modifier. 6_ le propriétaire saisit les informations du bien à modifier. 7_ le système confirme la modification des informations.
Scénario alternatif	6_1.le formulaire rempli est incorrect ou incomplet. 6_2 le système indique au Propriétaire que le formulaire est incorrect, et demande au Propriétaire que remplir le formulaire pour la deuxième fois. 6_3.reprise de l'enchaînement à partir l'étape 7.

Tableaux 3.4 : Description du Cas d'utilisation « modifier un bien »

3.2.1.3.3 .3 « supprimer un bien »

Cas d'utilisation	Gérer un bien
Acteur	Administrateur
But	La suppression d'un bien.
Pré_ condition	L'administrateur s'authentifie par le système.
Post_ condition	Confirmation de suppression d'un bien
Scénario nominal	1_ le système présente la liste des options. 2_ l' administrateur choisit l'option « Gérer un bien ». 3_ le système affiche la liste des biens. 4_ l' administrateur sélectionne un bien pour la suppression. 5_ le système demande au l'administrateur la confirmation de la suppression.

	6_1' administrateur confirme suppression.
--	---

Tableaux 3.5 : Description du Cas d'utilisation « supprimer un bien »

3.2.1.3.4 Description du Cas d'utilisation « recherche bien »

Cas d'utilisation	Recherche un bien
acteur	Client
but	Recherche dans les biens existants de la base de données du site web
Pré_condition	Néant
Post_condition	Afficher le résultat de la recherche
Scénario nominal	1-le client demande la recherche d'un bien 2-le système affiche les critères des recherches 3-le client choisit le critère de recherche 4- le système affiche la liste des biens selon le critère choisit s'ils existent
Scénario alternatif	4-1: si les biens n'existent pas le système affiche un message le système affiche le message. " le bien n'existe pas"

Tableaux 3.6 : Description du Cas d'utilisation « recherche bien »

3.2.1.3.5 Description du cas d'utilisation « Consulter une annonce »

Cas d'utilisation	Consulter une annonce.
Acteur	Client.
But	Consulter les informations des annonces existantes.
Pré condition	Néant
Post condition	Néant
Scénario nominal	1. Le client demande la liste des annonces existantes. 2. Le système envoi la liste des annonces pour consultation. 3. Le client consulte de la liste des annonces. 4. Le client choisit l'annonce. 5. Le système envoie les informations de l'annonce sélectionné.

Tableaux 3.7 : Description du cas d'utilisation « consulter une annonce »

3.2.1.3.6 Description du cas d utilisation « choisir une annonce »

Cas d'utilisation	Choisir une annonce
Acteur	Client

But	Choisit une annonce permet les existant
Pre_ condition	Le cas d'utilisation « consulter annonce » doit être exécuté
Post_ condition	Le système enregistre l'annonce
Scénario nominal	1-le client sélectionne une annonce 2-le système affiche un formulaire qui contient les informations du client 3- le client saisit les informations puis demande l'enregistrement de l'annonce 4-le système enregistre l'annonce
Scénario alternatif	3-1 :les informations sont erronées 3-2 :le système indique au client que les informations sont erronés et demande de les entrer une nouvelle fois 3-3 : reprise de l'enchainement à partir de l' étape 2-3

Tableaux 3.8 : Description du cas d'utilisation « choisir une annonce »

3.2.1.3.7 Description du cas d utilisation « gérer compte »

3.2.1.3.7.1« supprimer un compte »

Cas d'utilisation	Supprimer compte.
Acteur	Administrateur.
But	Permettes de supprimer d'un compte.
Pré_ condition	L'Administrateur s'authentifie par le système.
Post_ condition	Le compte va supprimer.
scénario nominal	1_ le système envoi la liste des comptes existants ; 2_ l' Administrateur sélectionne les comptes qu'il veut supprimer puis confirme la suppression. 3_ l' Administrateur valide la suppression.

Tableaux 3.9 : Description du Cas d'utilisation « Supprimer compte »

3.2.1.3.7.1« valider un compte »

Cas d'utilisation	Valider compte.
Acteur	Administrateur.
But	Permettes de valider d'un compte.
Pré_ condition	L'Administrateur s'authentifie par le système.
Post_ condition	Le compte va valider.
scénario nominal	1_ le système envoi la liste des comptes existants ; 2_ l' Administrateur sélectionne les comptes qu'il veut valider puis confirme la validation. 3_ l' Administrateur effectue la validation.

Tableaux 3.10 : Description du Cas d'utilisation « valider compte »

3.2.2 Les diagrammes de séquences système

3.2.2 .1 Diagramme de séquence Authentification

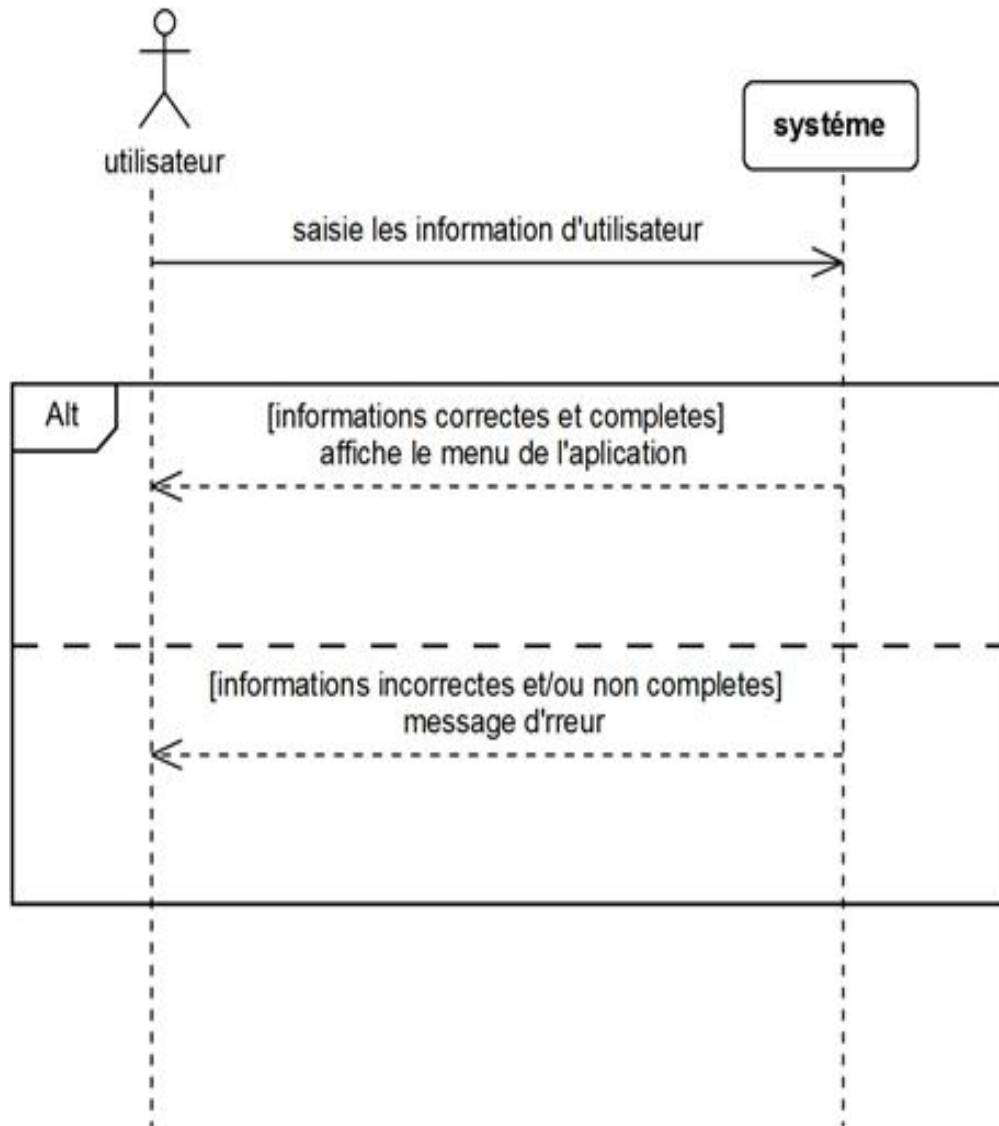


Figure3.1 : diagramme de séquence Authentification

3.2.2 .2 Diagramme de séquence créer un compte

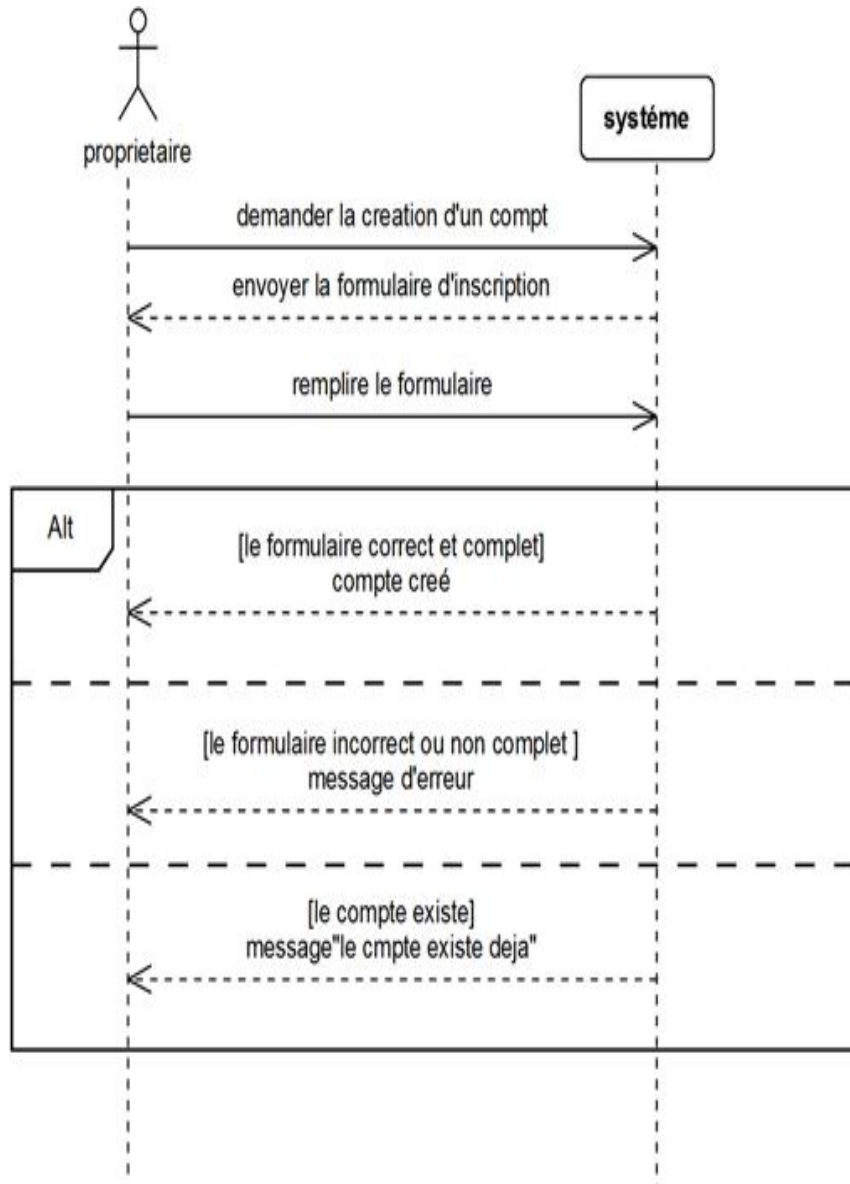


Figure3.2 : diagramme de séquence créer un compte

3.2.2 .3 Diagramme de séquence Gérer un bien

3.2.2 .3.1 proposer un bien

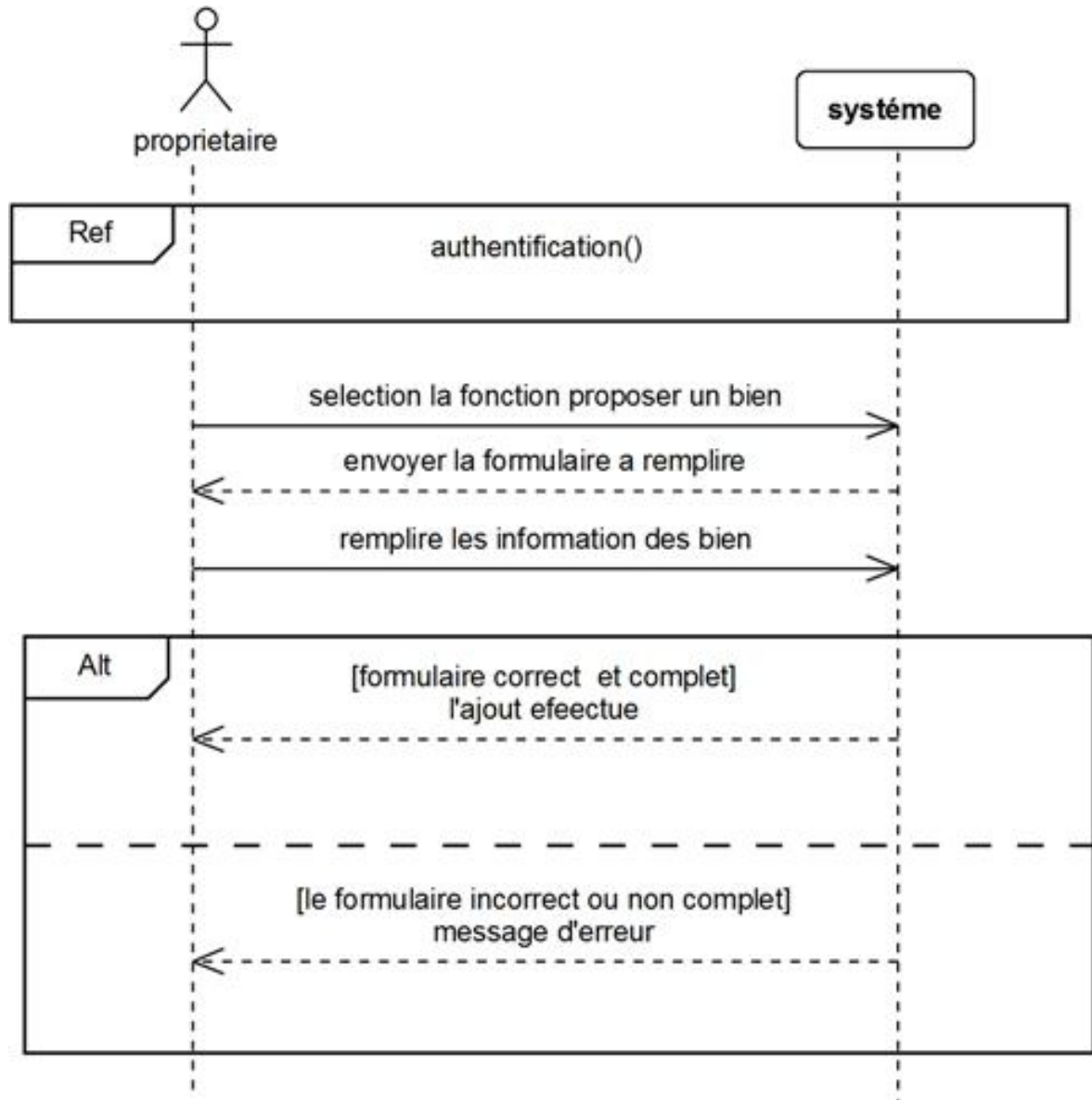


Figure3.3 : diagramme de séquence proposer un bien

3.2.2 .3.2 modifier un bien

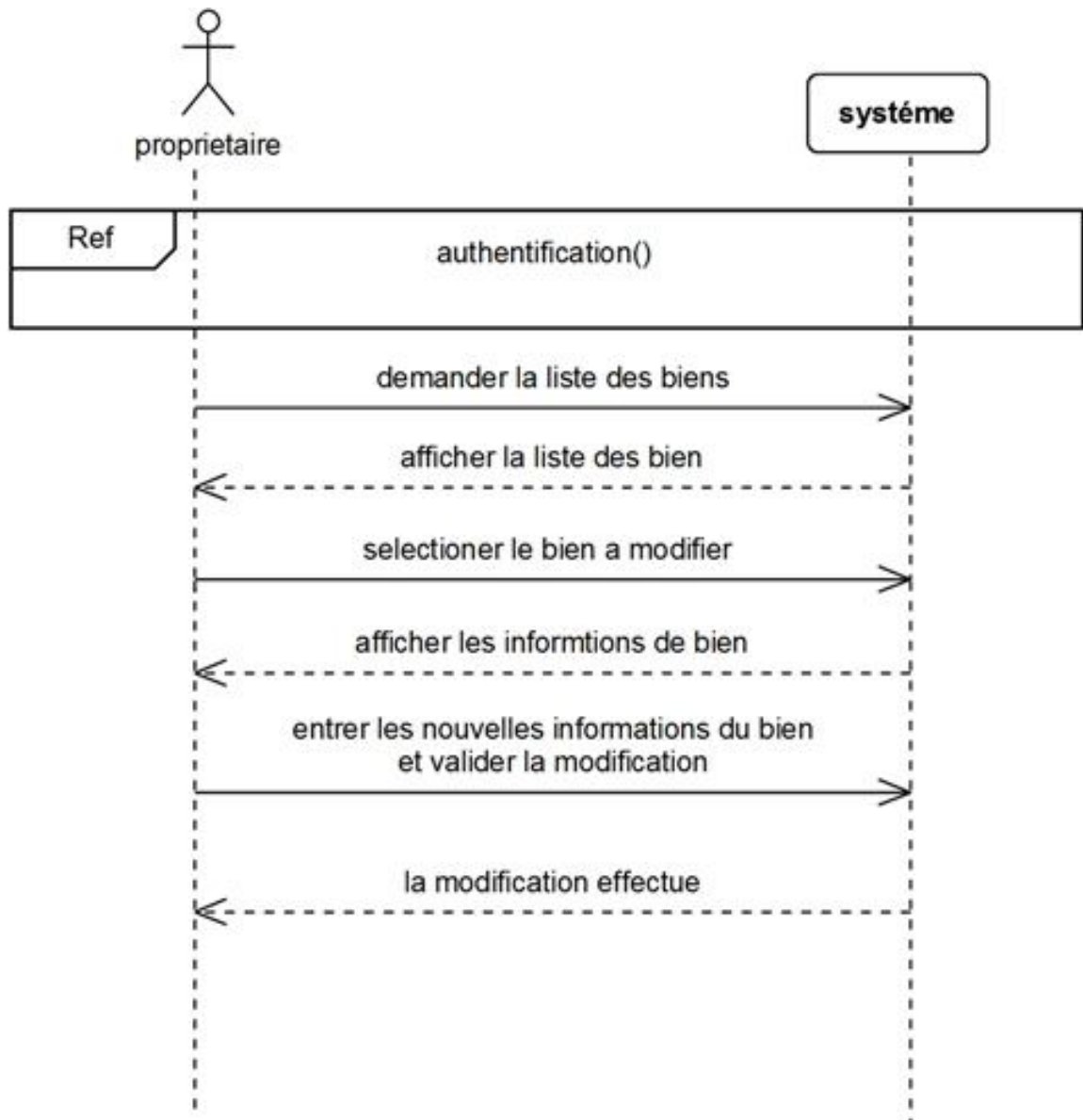


Figure3.4 : diagramme de séquence Gérer un bien (modifier)

3.2.2 .3.3 supprimer un bien

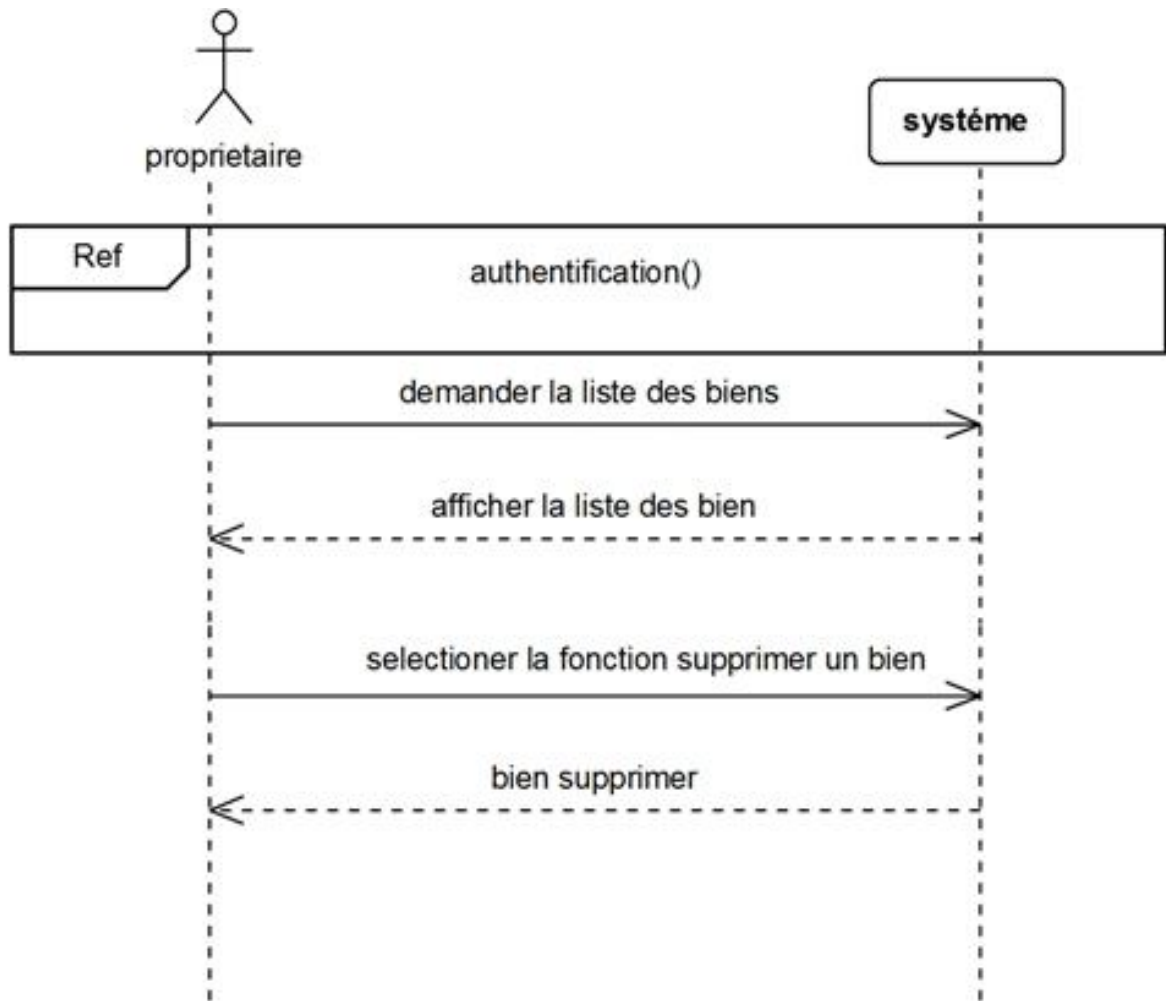


Figure3.5 : diagramme de séquence Gérer un bien (supprimer)

3.2.2 .4 Diagramme de séquence recherche un bien

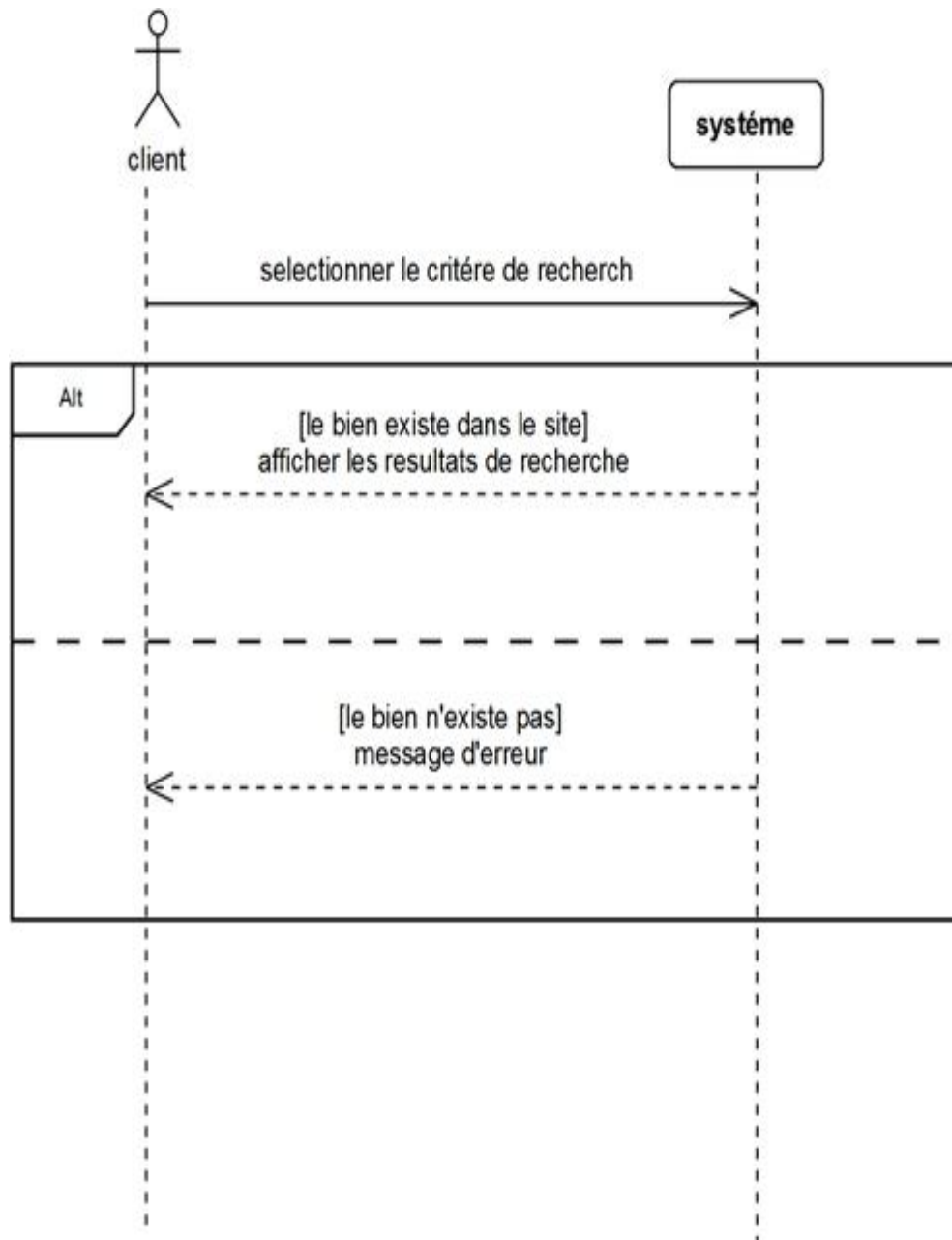


Figure3.6 : diagramme de séquence recherche un bien

3.2.2 .5 Diagramme de séquence consulter annonce

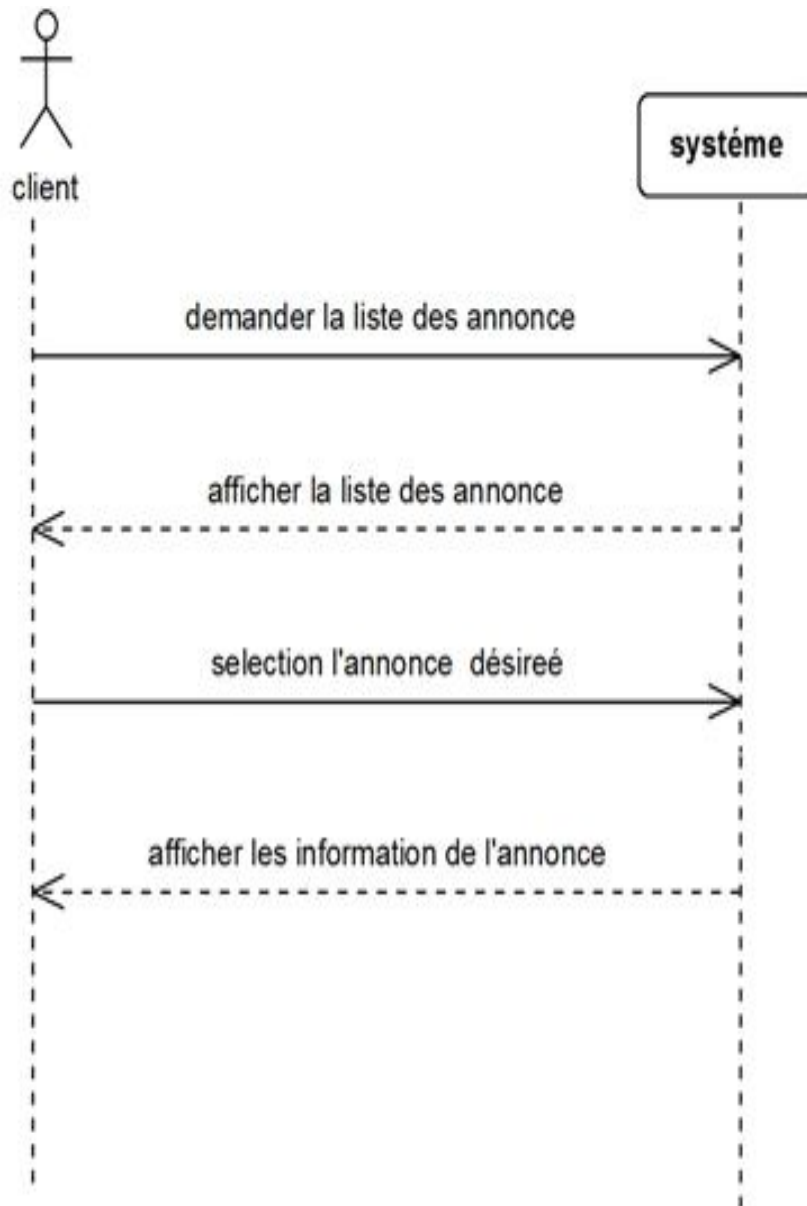
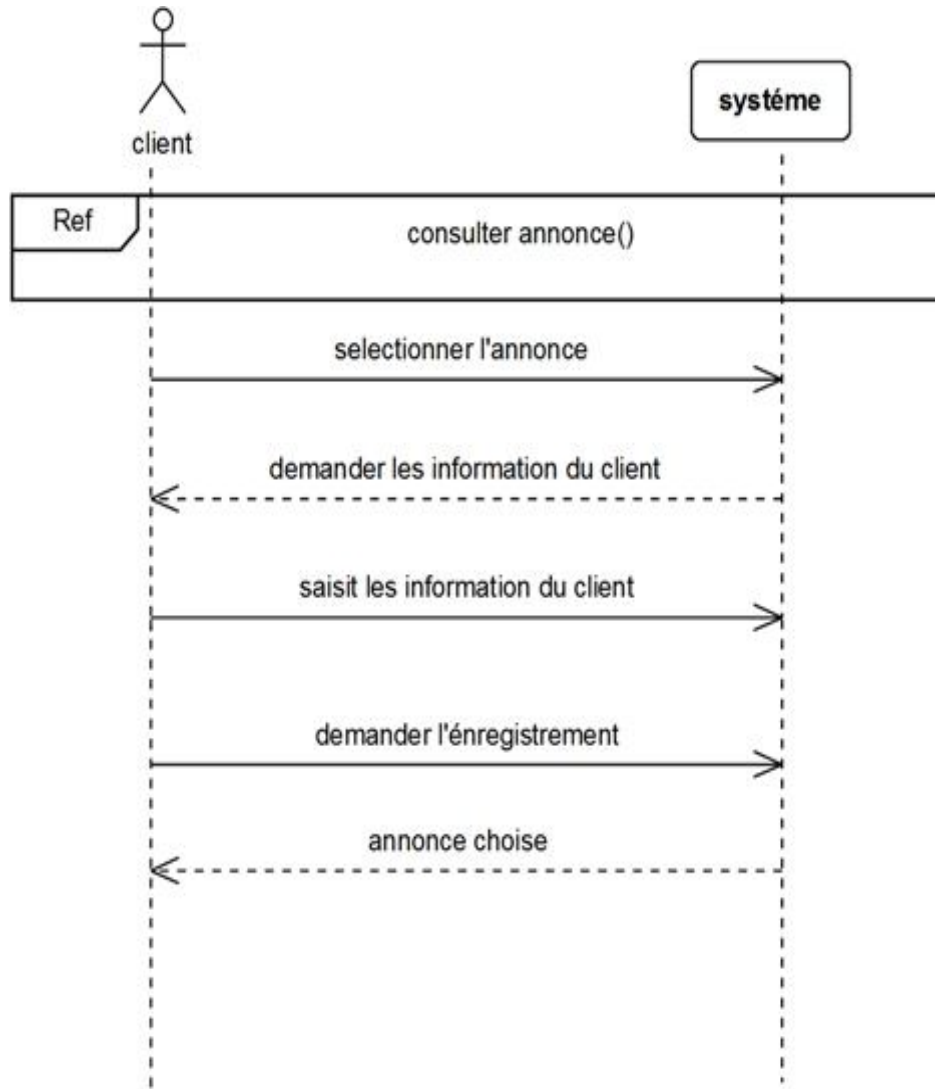


Figure3.7 : diagramme de séquence consulter annonce

3.2.2 .6 Diagramme de séquence choisir annonce**Figure3.8 : diagramme de séquence choisir annonce**

3.2.2 .7 Diagramme de séquence Gérer compte

3.2.2 .7.1 supprimer un compte

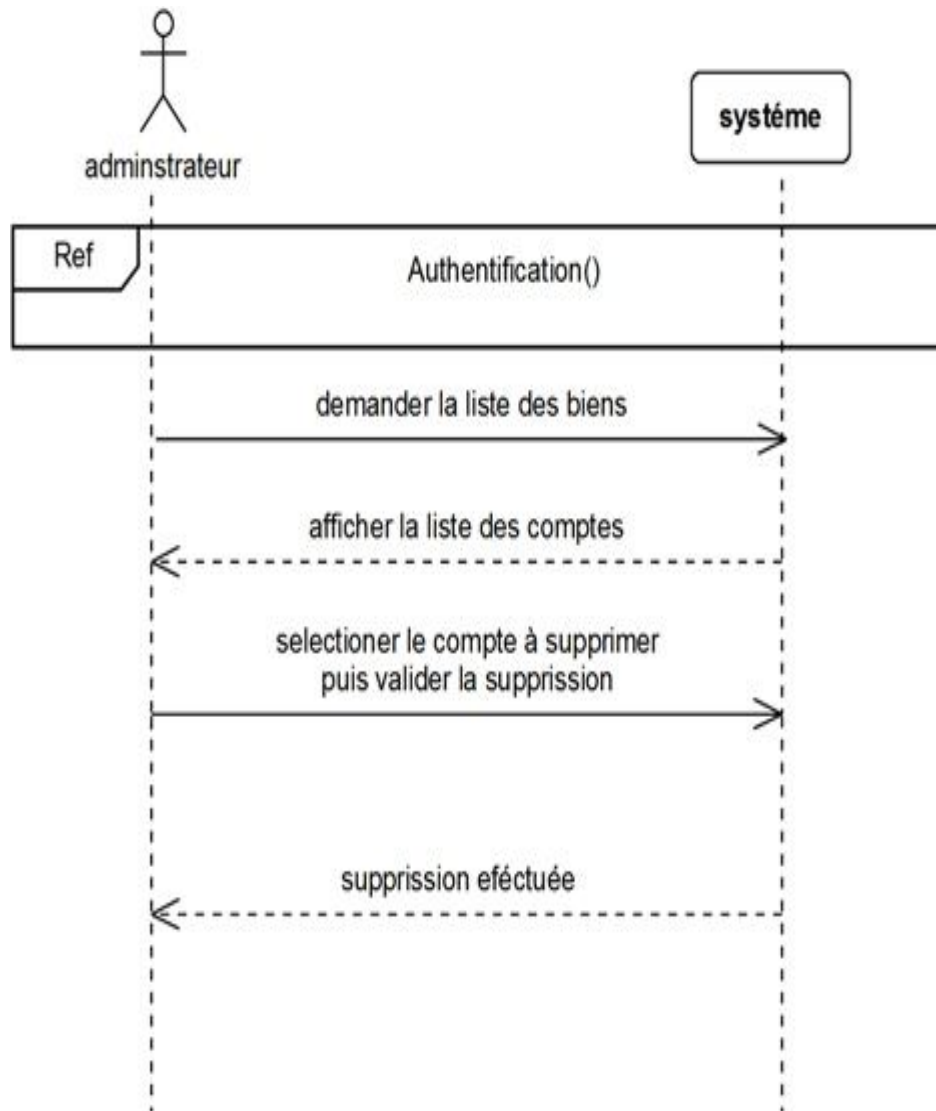


Figure3.9 : diagramme de séquence Gérer d'un compte (supprimer)

3.2.2 .7.2 valider un compte

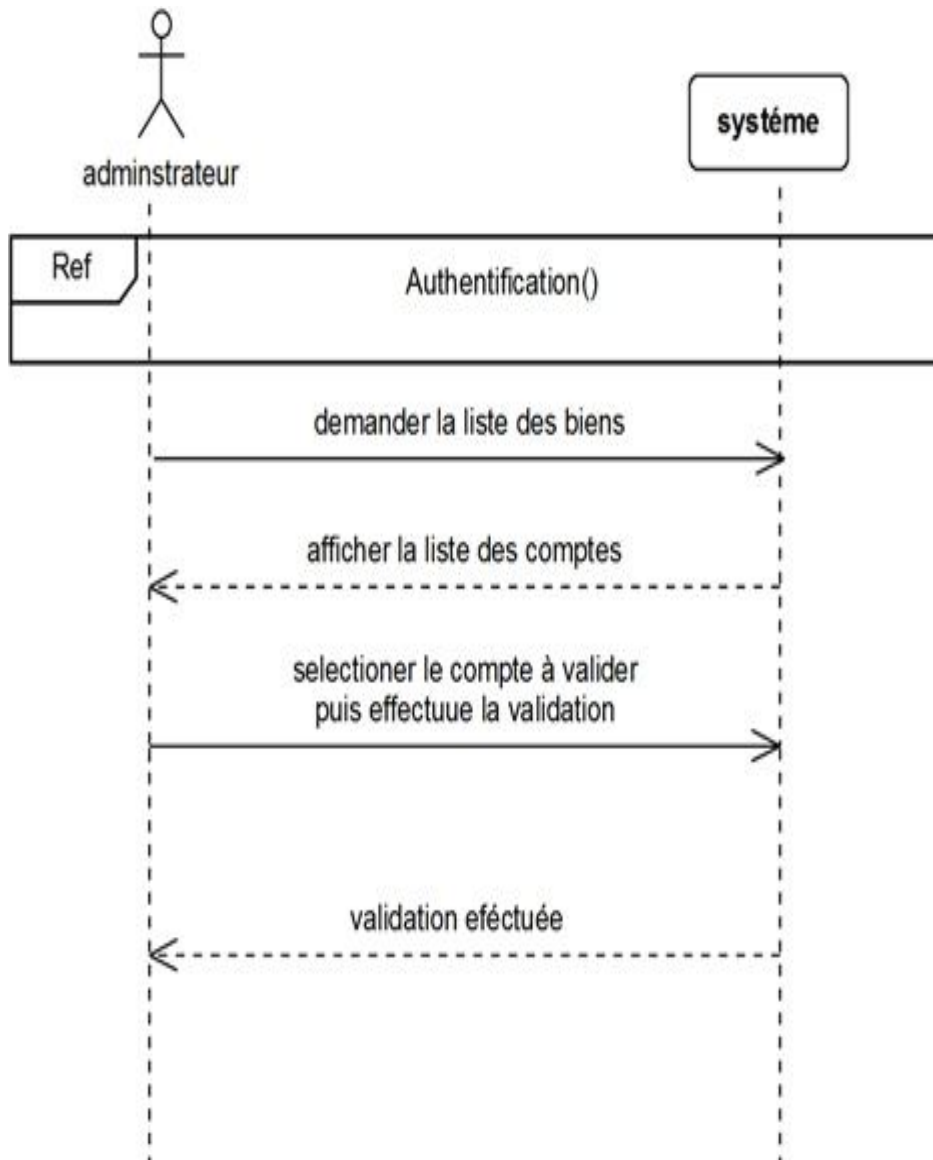


Figure3.10 : diagramme de séquence Gérer d'un compte (valider)

3.3 La phase d'analyse :

3.3.1 : Modèle du domaine

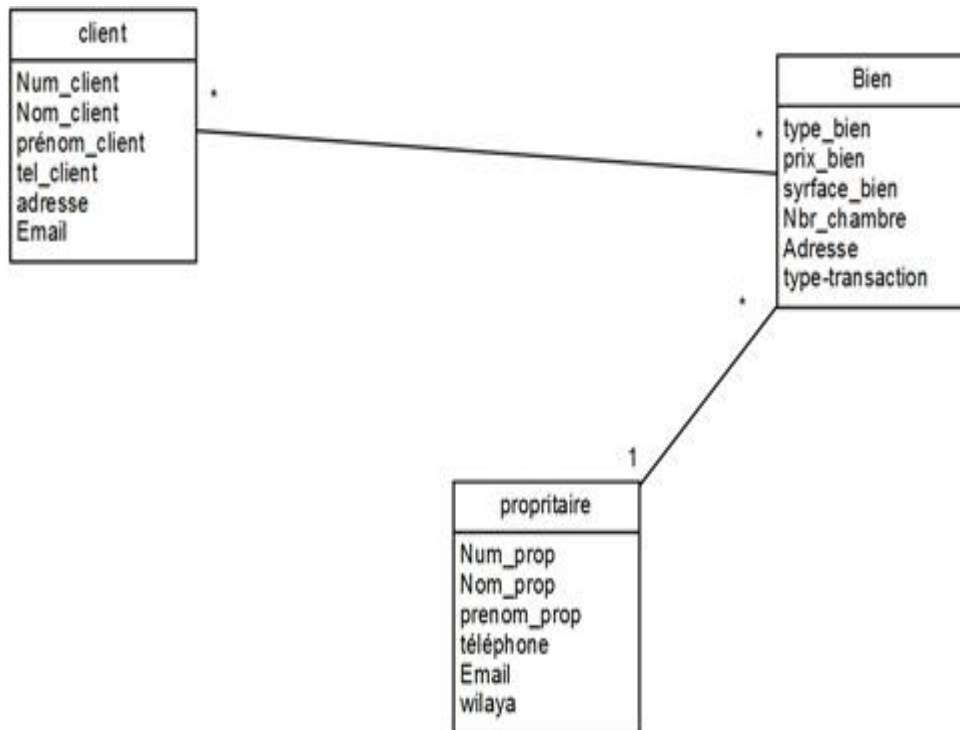


Figure3.11 : diagramme de class

3.3.2 : diagramme de class participantes

3.3.2.1 Authentification

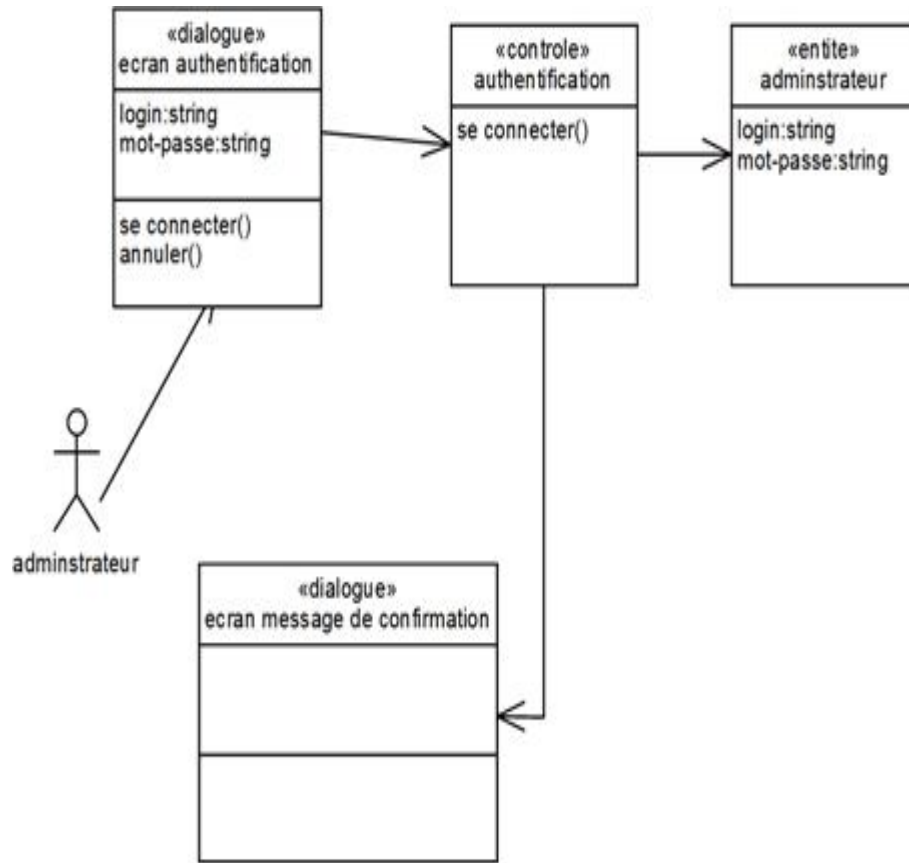


Figure3.12 : diagramme de class participant Authentification

3.3.2 .2 Gérer un bien

3.3.2 .2.1 proposer un bien

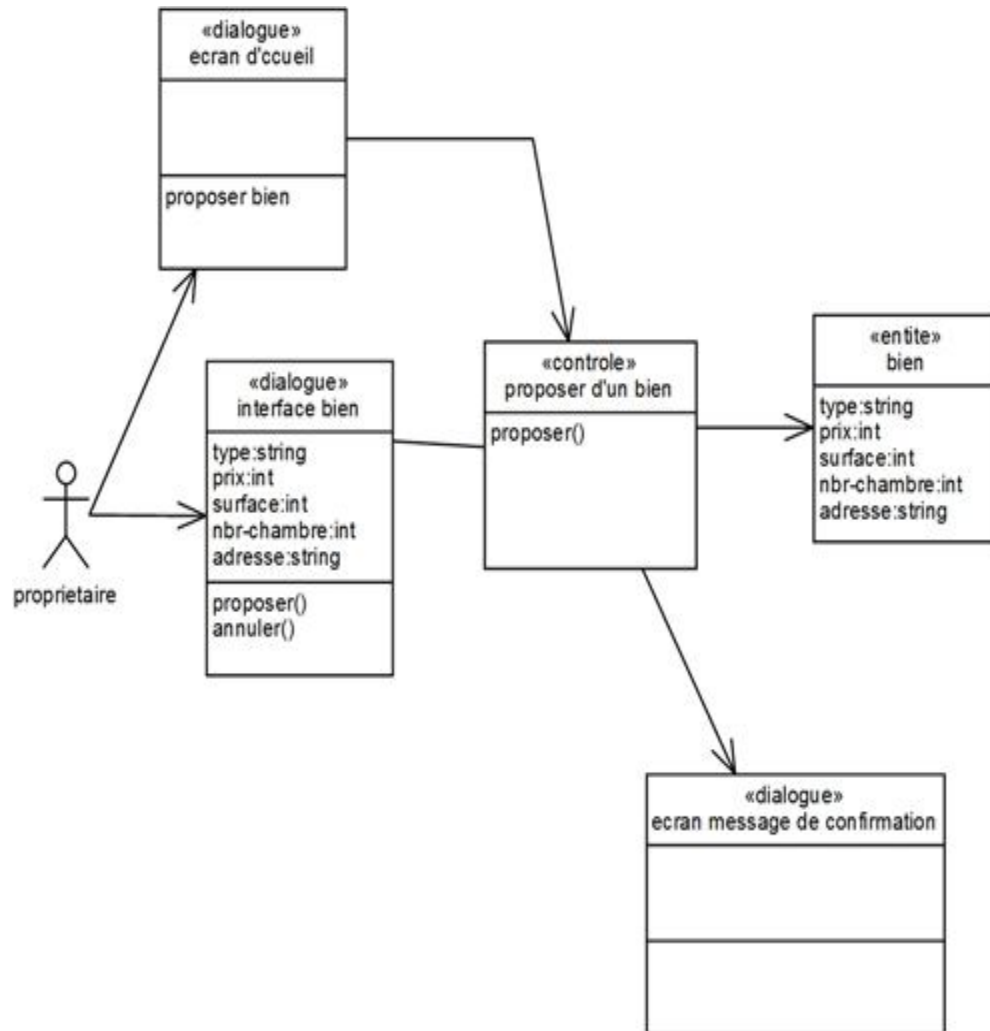


Figure3.13 : diagramme de class participant proposer un bien

3.3.2 .2 .2 modifier un bien

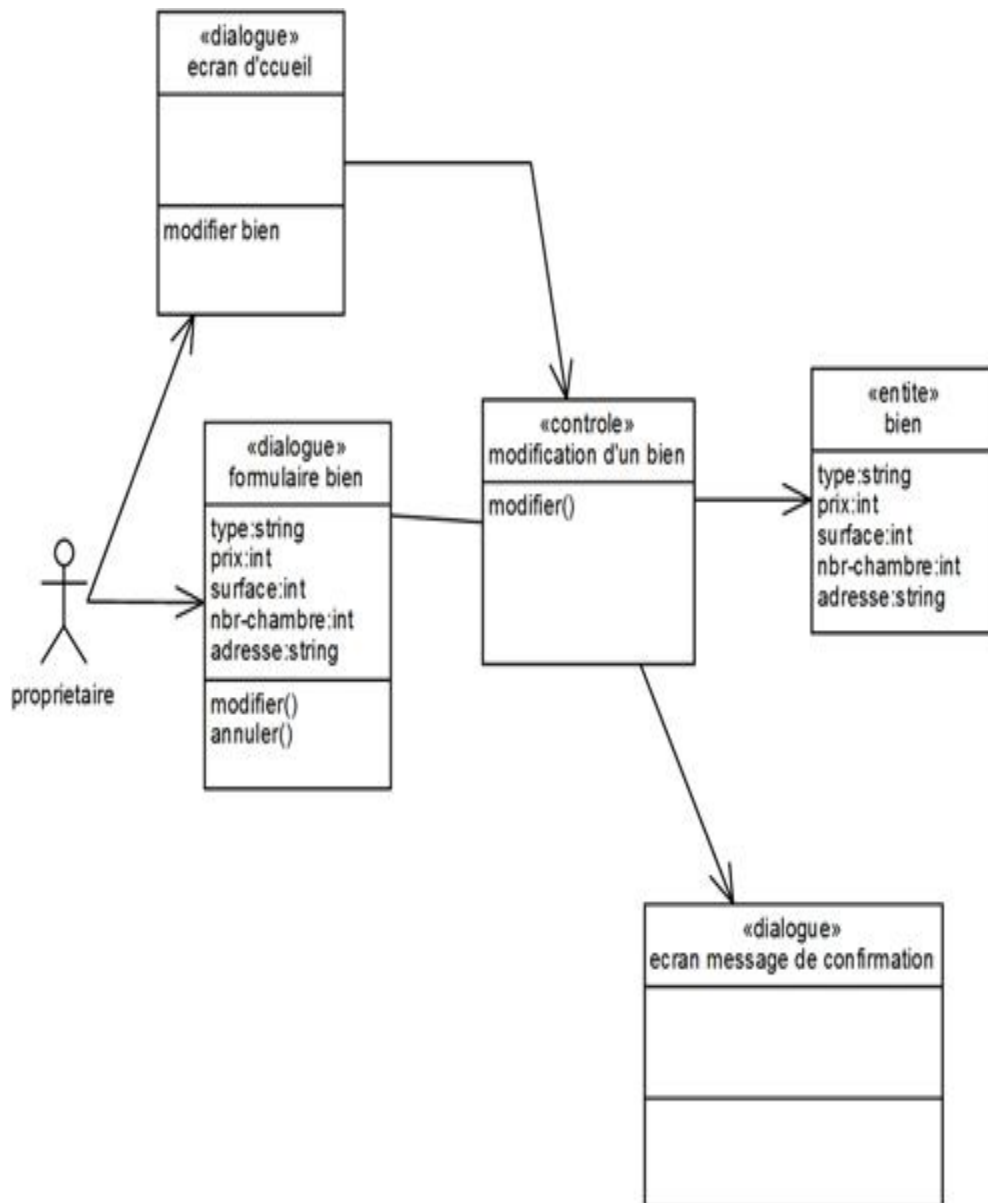


Figure3.14 : diagramme de class participant modifier un bien

3.3.2 .2 .3 supprimer un bien

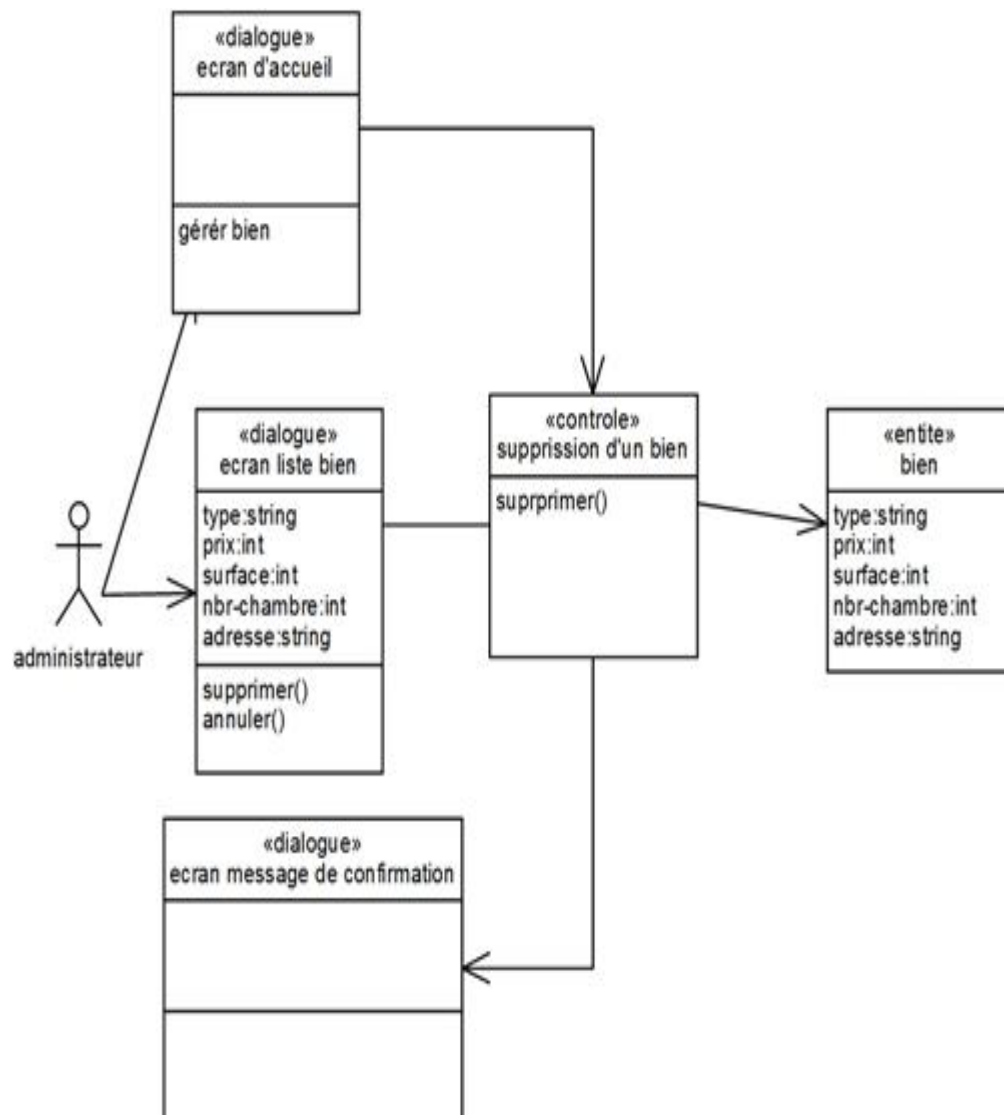


Figure3.15 : diagramme de class participant supprimer un bien

3.3.2 .3 consulter annonce

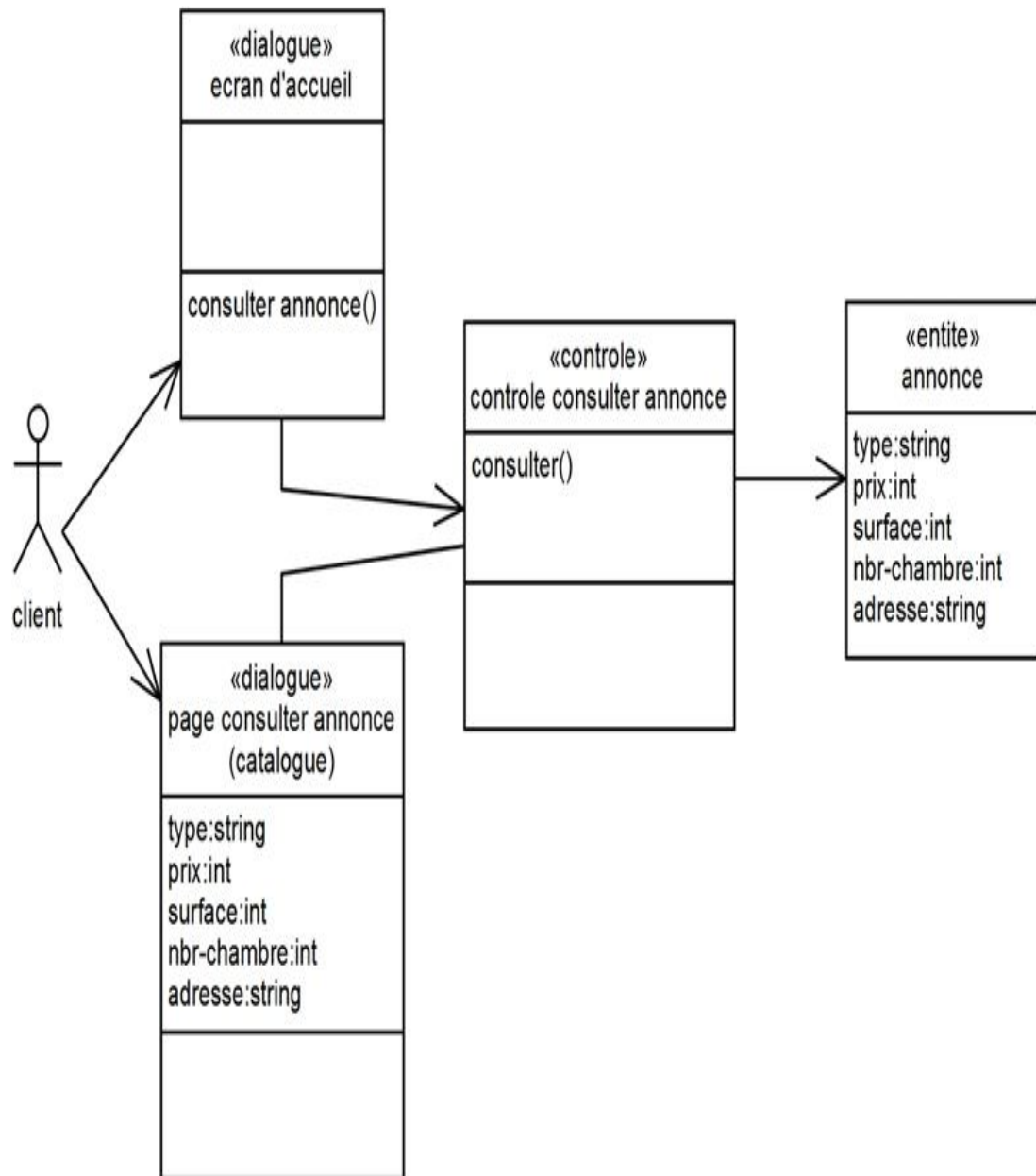


Figure3.16 : diagramme de class participant consulter annonce

3.3.2 .4 utiliser annonce

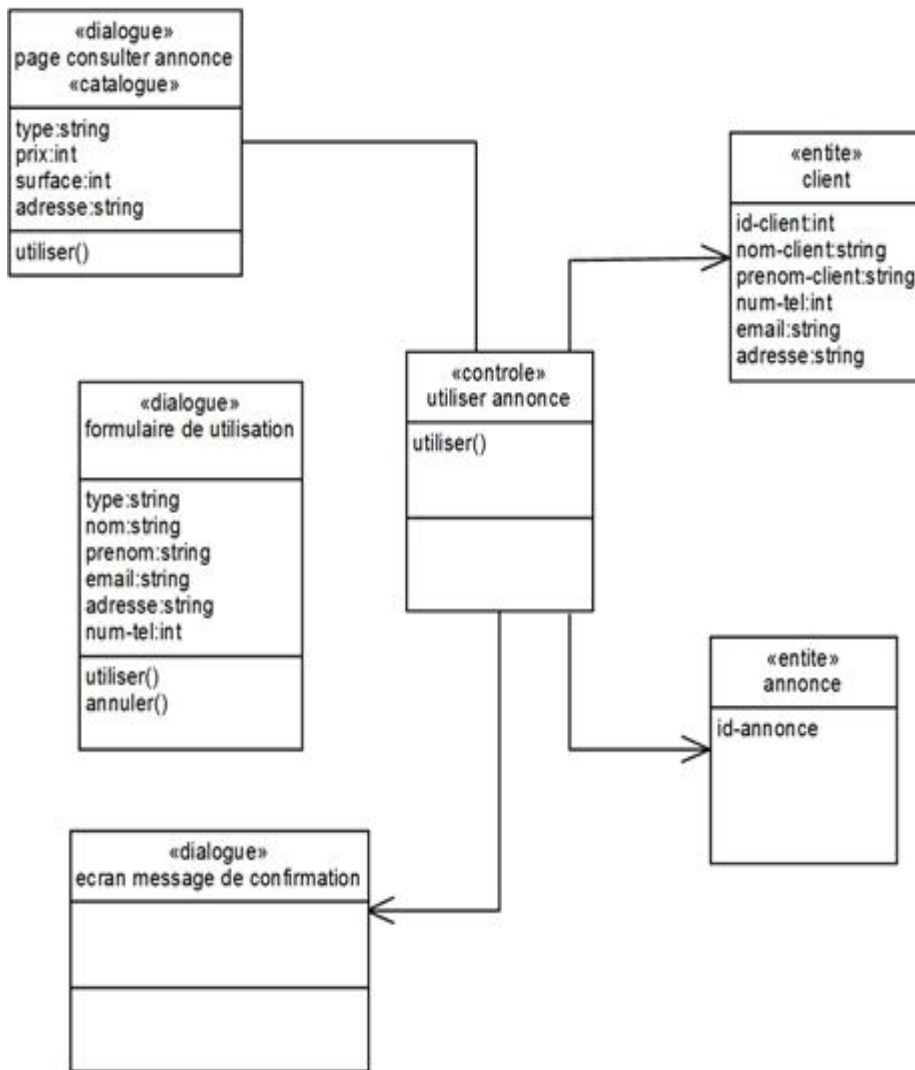


Figure3.17 : diagramme de class participant choisir annonce

3.3.3 Diagrammes d'activités

3.3.3.1 Authentification

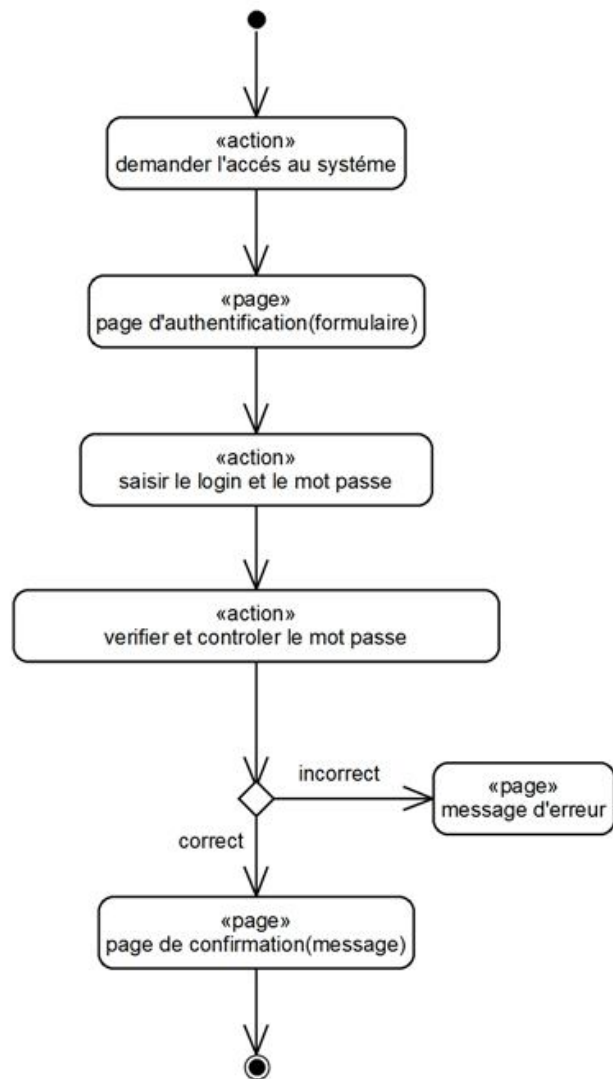


Figure3.18 : diagramme d'activité Authentification

3.3.3 .2 Gérer un bien

3.3.3 .2.1 proposer un bien

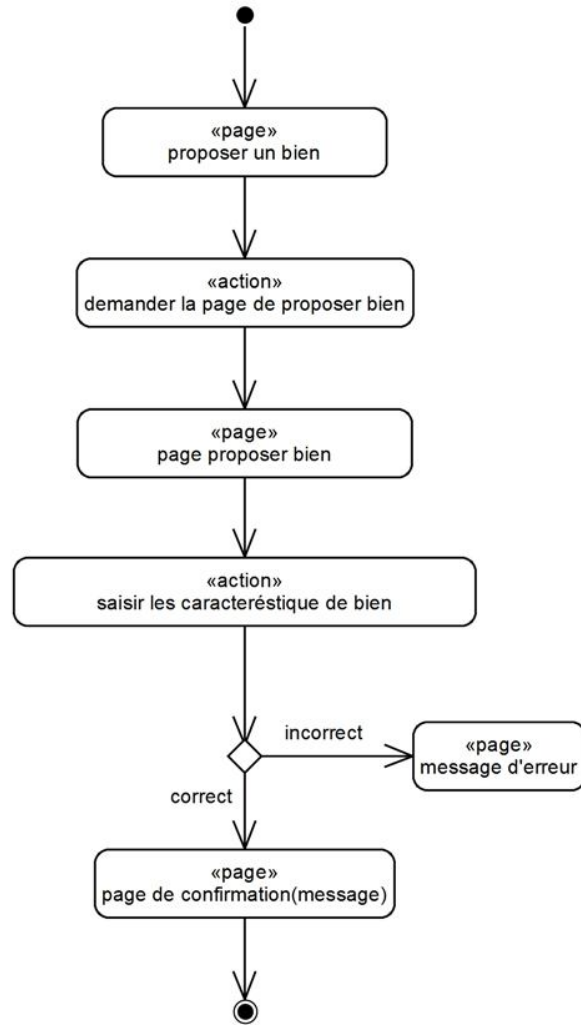


Figure3.19 : diagramme d'activité proposer un bien

3.3.3 .2 .2 modifier un bien

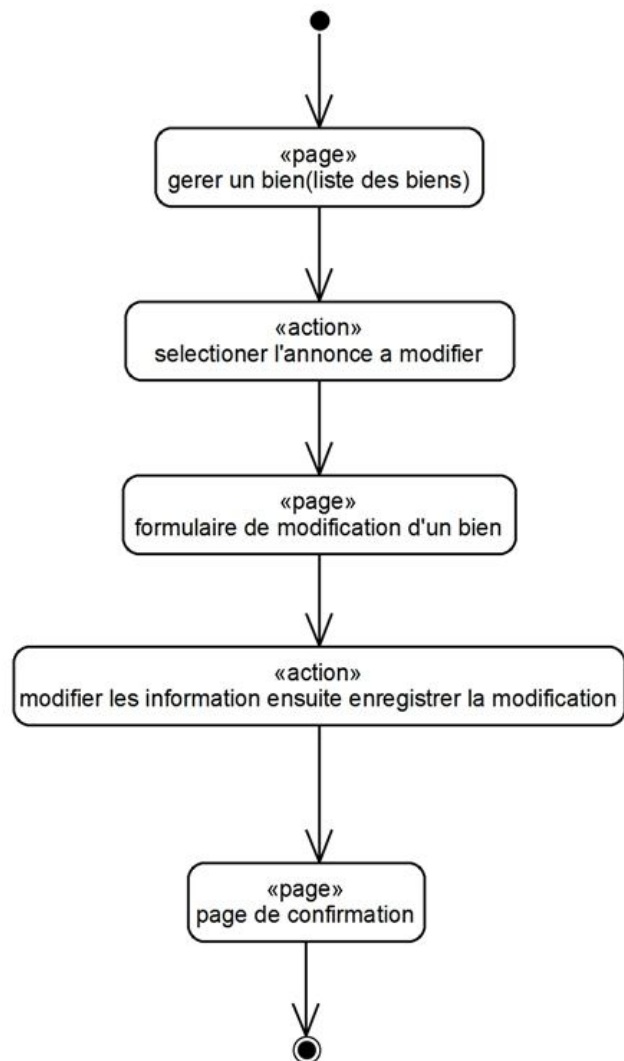
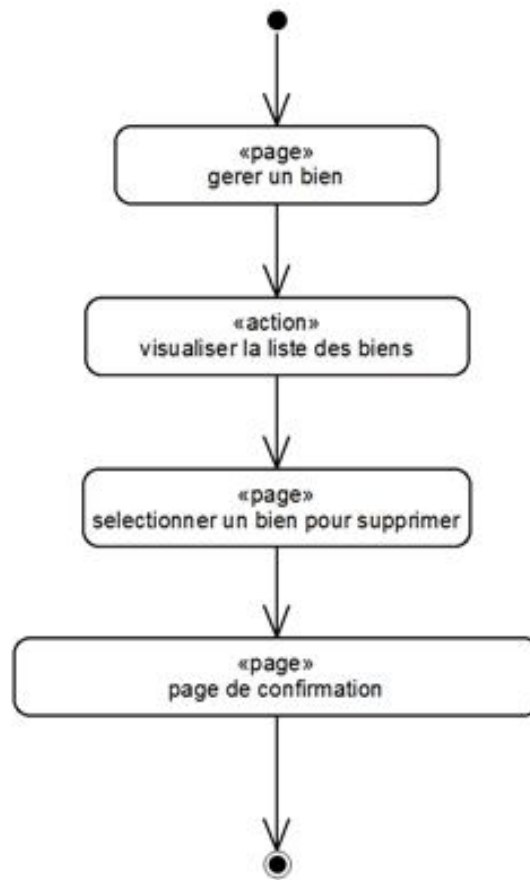
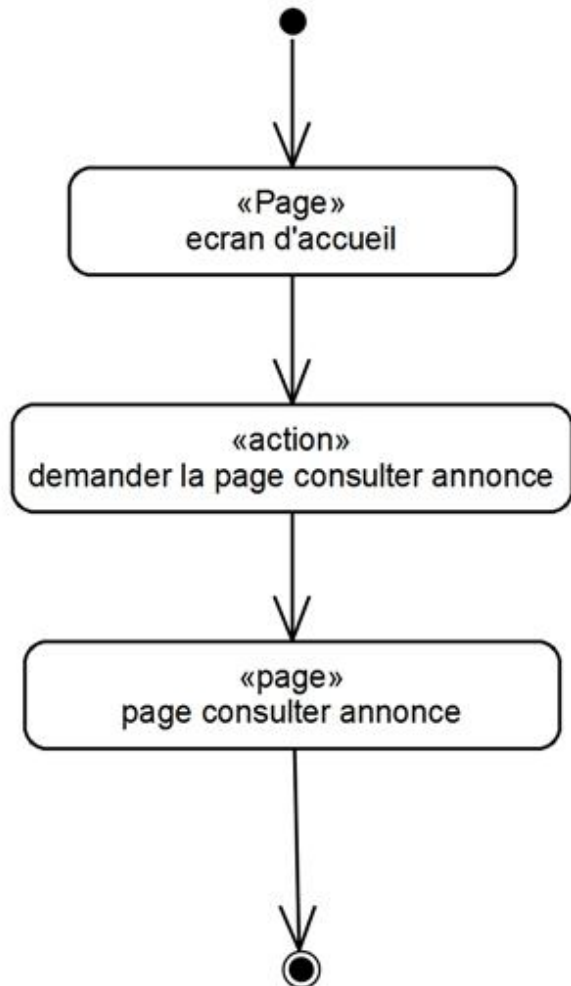


Figure3.20 : diagramme d'activité modifier un bien

3.3.3 .2 .3 supprimer un bien**Figure3.21 : diagramme d'activité supprimer un bien**

3.3.3 .3 consulter annonce**Figure3.22 : diagramme d'activité consulter annonce**

3.3.3 .4 choisir une annonce

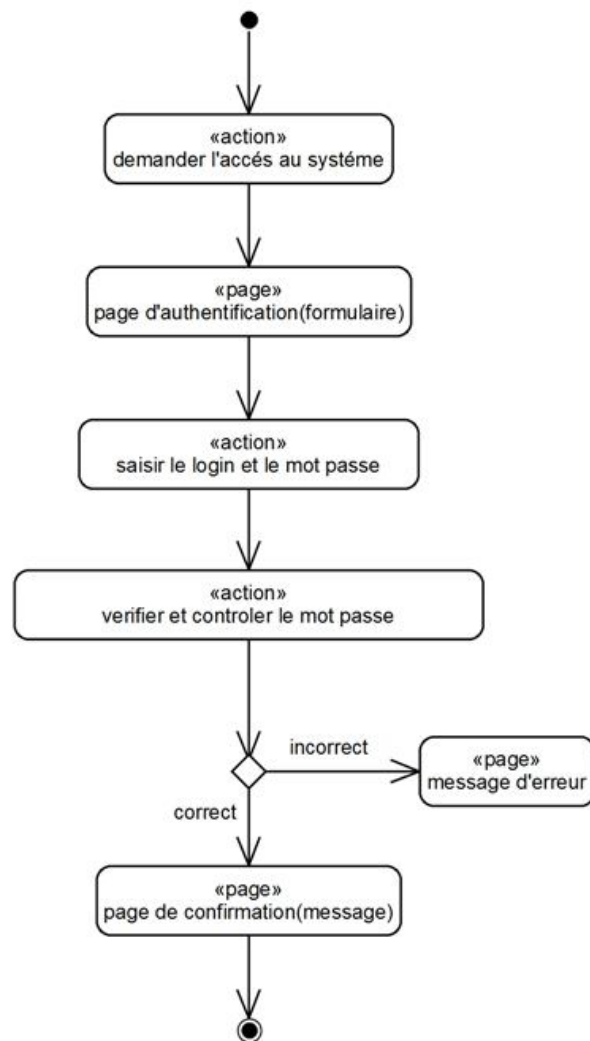


Figure3.23 : diagramme d'activité choisir une annonce

3.4 Conclusion :

Dans cette étape nous avons exprimé clairement les objectifs attendus du futur système concevoir, ainsi que l'analyse associée à chaque cas d'utilisation. Dans le chapitre suivant nous allons détailler la prochaine étape « l'implémentation ».

**CHAPITRE 4 :
IMPLIMENTATION**

4.1 Introduction

La réalisation d'un site web dynamique a besoin de plusieurs logiciels de développements. Pour cela, nous avons choisi quelques outils et langages parmi ceux existants et qui répondent à nos besoins.

Dans ce chapitre, nous allons citer la liste des outils utilisés pour la réalisation du site tout en expliquant le rôle de chacun.

4.2 Les Langages de programmation

4.2.1 Le langage HTML

HTML est l'abréviation de HyperText Mark up Langage. Il a été présenté en version 1 en 1993. Au moment de l'apparition de MOSAIC premier navigateur WWW. Depuis lors, HTML a été largement amélioré et étendu. Le langage HTML permet de créer des documents indépendants de toute plateforme, bien adaptés à des échanges d'information dans un environnement hétérogène comme le Web.

Les pages HTML ont toutes la même structure de base. Elles se composent de plusieurs balises (tags) ou instruction, placées entre les caractères < et >. La plupart d'entre elles vont de pair avec une balise de fin, servant à interrompre leur action. La balise de fin est identique à celle de départ sauf que l'instruction proprement dite est précédée d'un slash (/). Ainsi, un document HTML commence par <html> et se termine par </html>. Entre les balises se trouve le texte définissant le contenu ou le comportement (script) du document. Même si rien n'est inséré, une page HTML comporte néanmoins quelques éléments il s'agit des balises qui représentent la structure de base. [8]

4.2.2 Java Script

JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives. C'est un langage orienté objets à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés.

Ce Langage de programmation est développé par Sun, inspiré de C++. Les programmes Java peuvent être appelés depuis des documents HTML ou de manière autonome. Lorsqu'ils s'exécutent à partir d'une page Web, on les appelle des applets Java. Lorsqu'ils s'exécutent sur un serveur Web, on les dénomme Servlet1. [9]

4.2.3 Le langage PHP (Personnel Home Page)

Le PHP, Personnel Home Page ou HyperText préprocesseur, est un langage de programmation Web. Le code *PHP* est directement inclus dans les pages Hypertexte,

il doit être placé entre les balises « < ? PHP » et « ? > ». On appelle ce genre de langage, un langage de script. [10]

PHP permet de créer des pages interactives. Une page interactive permet à un visiteur de saisir des données personnelles. Ces dernières sont ensuite transmises au serveur, où elles peuvent rester stockées dans une base de données pour être diffusées vers d'autres utilisateurs. Un visiteur peut, par exemple, s'enregistrer et retrouver une page adaptée à ses besoins lors d'une visite ultérieure. Il peut aussi envoyer des e-mails et des fichiers sans avoir à passer par son logiciel de messagerie.

En associant toutes ces caractéristiques, il est possible de créer aussi bien des sites de diffusion et de collecte d'information que des sites d'e-commerce, de rencontres ou des blogs.

PHP est un langage de programmation, très proche du langage C dont il reprend l'essentiel de la syntaxe et destiné à être intégré dans des pages HTML. Contrairement à d'autres langages,

PHP est exclusivement dédié à la production de pages HTML générées dynamiquement. Les codes du PHP sont appelés « scripts », et ils sont inclus dans le code HTML.

A l'origine du *PHP*, il y a les CGI (Common Gateway Interface) et plus particulièrement des langages tels que le PERL ou C. On trouve dans le *PHP* un grand nombre de similitudes avec ces derniers et plus particulièrement au niveau de la syntaxe. Généralement, Le *PHP* est caractérisé par le fait qu'il est:

- **Un langage polyvalent** : PHP s'inscrit dans le mode des logiciels libres, cela signifie que la communauté des programmeurs qui s'axent autour du *PHP* participe activement au développement de ce langage.
- **un langage Simple** : PHP propose un langage et un modèle de développement très simple. La vocation historique de PHP était de permettre à n'importe quel informaticien de développer rapidement et sans formation pointue préalable une application web dynamique. Elle s'avère tellement simple que tous les hébergeurs grand public ont retenu PHP.
- **Un langage exécuté coté serveur** : Lorsqu'on effectue une requête sur une page écrite en *PHP*, celle-ci est interprétée et exécutée par le serveur. Le résultat qui apparaît au client est totalement dépourvu de code et celui-ci reste confidentiel.
- **Un langage très simple à assimiler** : La programmation en *PHP* est très libre, pas besoin, par exemple, de déclarer les variables ou redimensionner les tableaux manuellement...etc.
- **Un langage de script** : Le *PHP* n'est pas compilé à chaque fois qu'une requête est effectué sur une page *PHP*, le serveur interprète celui-ci et le traduit en langage machine exécutable.

4.2.4 MYSQL

Le langage SQL est un langage normalisé de gestion de base de données. MYSQL, le plus populaire des serveurs de bases de données SQL Open Source, est développé, distribué et supporté par MYSQL AB (MYSQL AB est une société commerciale, fondée par les développeurs de MySQL, qui développent leur activité en fournissant des services autour de MySQL).

De nombreux fournisseurs d'accès Internet, et hébergeurs de sites proposent un accès gratuit à une plate-forme PHP/MySQL, ceci soutient le choix de MySQL. D'autres facteurs d'ordre technique existent :

- **Vitesse de traitement** : MySQL se distingue par une vitesse de traitement accrue résultant de la réécriture de routine ISMA de gestion des entrées/sorties, et de l'adoption d'algorithmes d'appariement de tables optimisés et organisés autour de threads.
- **Fiabilité** : Des listes de diffusion active contribuent à la correction rapide d'éventuels bogues de MySQL. De plus, des utilitaires de teste de cohérence automatique existent, comme MYIAMCHK.
- **Compatibilité SQL** : MySQL est compatible avec SQL, il respecte la norme ANSI SQL 92.
- **Sécurité** : L'administrateur d'une base de données contrôle finement les catégories de personnes et les machines autorisées à se connecter, et l'authentification repose sur des mots de passe cryptés. [11]

Pour afficher le contenu d'une base de données, il faut faire une connexion au serveur de cette base de données. Pour ce faire, nous faisons appel à la commande suivante :

```
$connexion = mysql_pconnect("localhost","root","");
```

\$connexion: variable dans laquelle est stockée le «handle» de la base. C'est le ticket qui nous permettra de travailler avec la base.

Localhost: c'est le nom attaché à l'adresse IP 127.0.0.1. Dans d'autres cas, il faut remplacer localhost par l'adresse du serveur SQL.

Root: c'est l'utilisateur avec lequel nous nous connectons. Le root tous les droits, il est donc dangereux de l'utiliser sur un serveur distant. Généralement, il faut utiliser l'identifiant et le mot de passe.

4.3 Les outils de développement

4.3.1 UML diagramme

UML Diagramme est un programme qui fournit un ensemble complet d'outils de modélisation graphique, d'analyse et de conception dans le développement de logiciels basés sur les modèles UML, COM, OMT.

Nous avons utilisé cet outil pour produire les diagrammes UML que nous avons besoin pendant la modélisation de notre application.

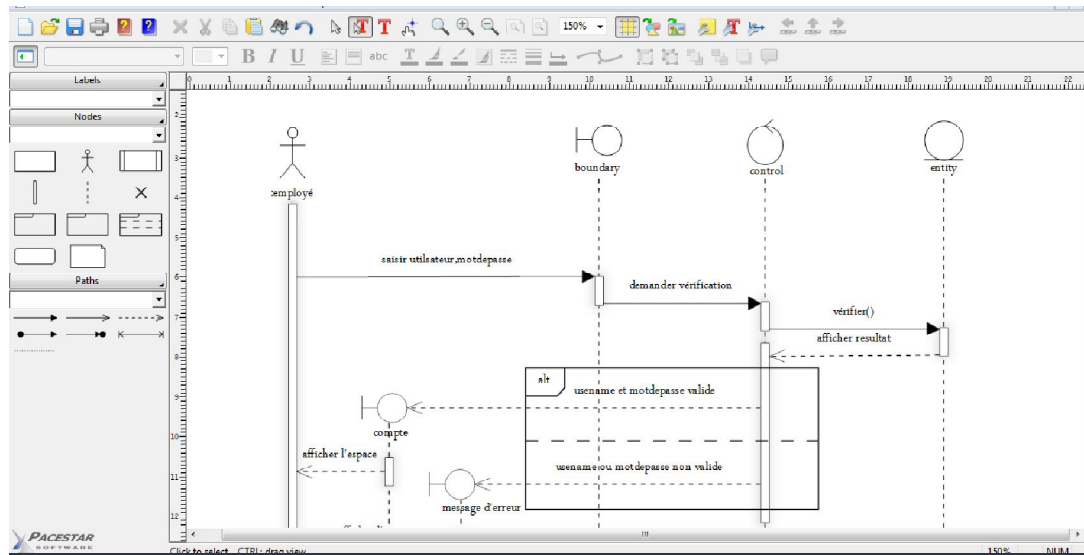


Figure 4.1 UML diagramme version:6.13

4.3.2 PH PMY Admin

PhpMyAdmin est une interface conviviale qui permet de gérer très facilement une base de données, sans nécessiter d'une connaissance avancée des requêtes SQL. Elle utilise conjointement une base MySQL et un moteur de scripts PHP. Cette interface peut donc fonctionner directement sur le serveur Web et être accessible par le biais d'un simple navigateur. Elle permette la:

- gestion des bases de données (Créer, copier, supprimer et modifier des tables);
- gestion des utilisateurs et leurs droits d'accès ;
- exécution des requêtes SQL.

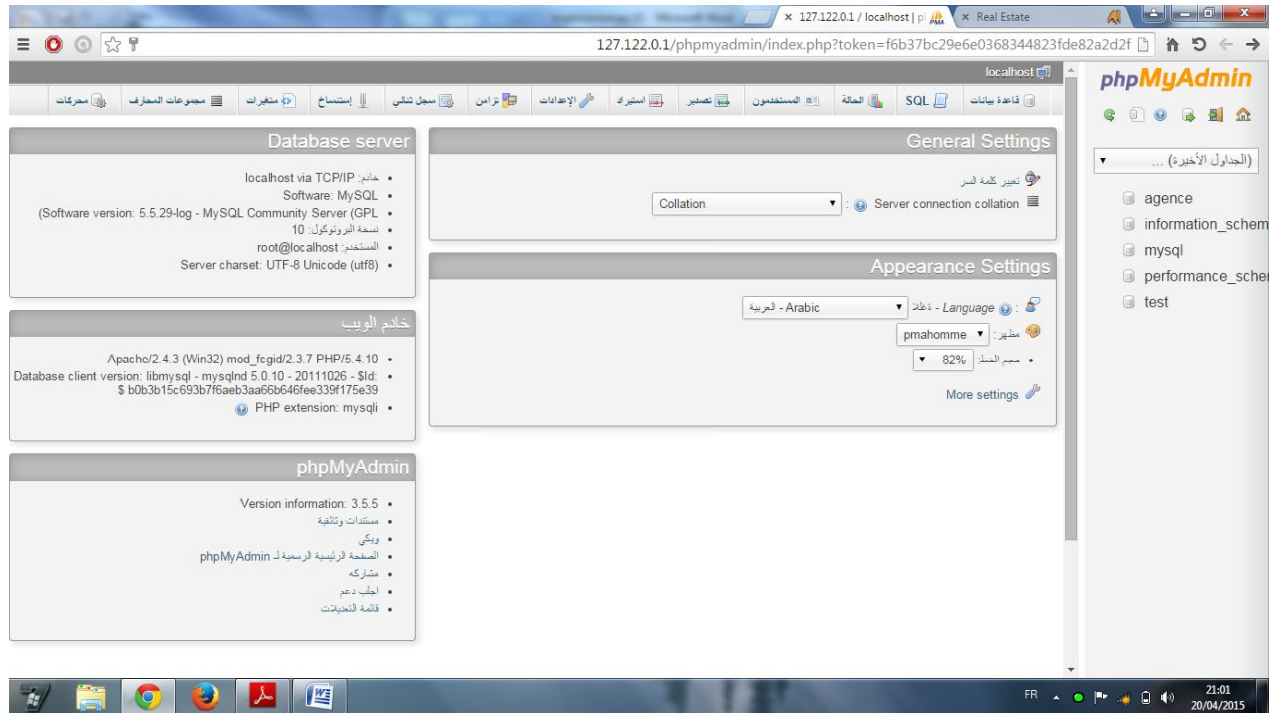


Figure 4.2: PHP MY Admin

4.4 Le passage du diagramme de classe au modèle relationnel

Nous donne ci –après quatre règles de (R1 à R4) pour traduire le modèle du domaine en un schéma relationnel équivalent. Il existe d'autre solution de transformation mais ces règles sont les plus simples et les plus opérationnelles :[7]

1) Transformation des entités /classes

La règle est simple **R1** :

* Chaque entité devient une relation, identifiant de l'entité devient clé primaire pour la relation.

* chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle identifiant.

2) Transformation des associations

Les règles de transformation que nous allons voir dépendent des cardinalités / multiplicités maximale des associations. Nous distinguons trois familles d'association:

a. Association 1 vers plusieurs (1...*) : La règle est la suivante :

R2 : Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

b. Association plusieurs vers plusieurs (*...*) : La règle et la suivante :

R3 : association/classe-association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des identités connecté à l’association. Chaque attribut devient clé étrangère si entité/classe connectée dont il devient une relation en vertu de la règle **R1**.

Les attributs d’association/classe-association doivent être ajoutés à la nouvelle relation. Ces attribut ne sont ni clé primaire, ni clé étrangère.

c . Association 1 vers 1 (1...1) : La règle et la suivante :

R4 : il faut ajouter un attribut de type clé étrangère dans la relation dérivée de l’entité ayant la cardinalité minimale égale à zéro. Dans le cas de diagramme UML il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un.

L’attribut porte le nom de la clé primaire de la relation dérivée d’entité classe connectée à l’association. Si les deux cardinalités minimales égales à zéro, le choix est donné entre les deux relations dérivées de la R1.

Si les deux cardinalités minimales égales à un, il est préférable de fusionner les deux entités/classe en une seule.

4.5 Implémentation de la base de données

Pour implémenter notre base des données baptisée « Agence», nous avons utilisé PHP MyAdmin et le système de gestion de base des Donnée MySQL. La figure ci-dessous présente notre base de données.

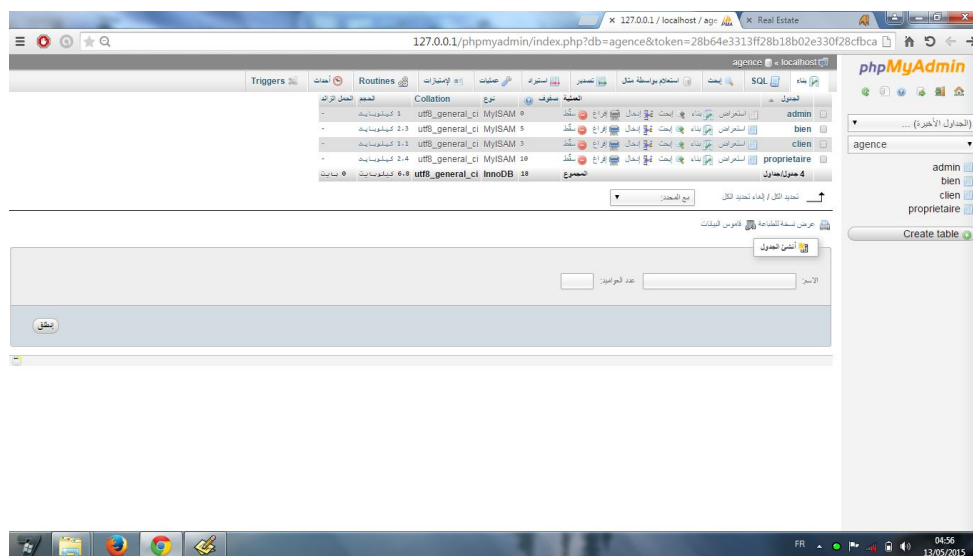


Figure 4.3 : structure de base de données.

Administrateur (username, password)

bien(id bien,type, surface, prix, telephone, wilaya, pieces, chambres, #nump)

proprietaire(nump, nom, prenom, wilaya, email, password, sexual ,)

clien(nom, prenom, telephone, adresse, email)

4.6 Présentation de quelques interfaces de l'application

4.6.1 page d'accueil

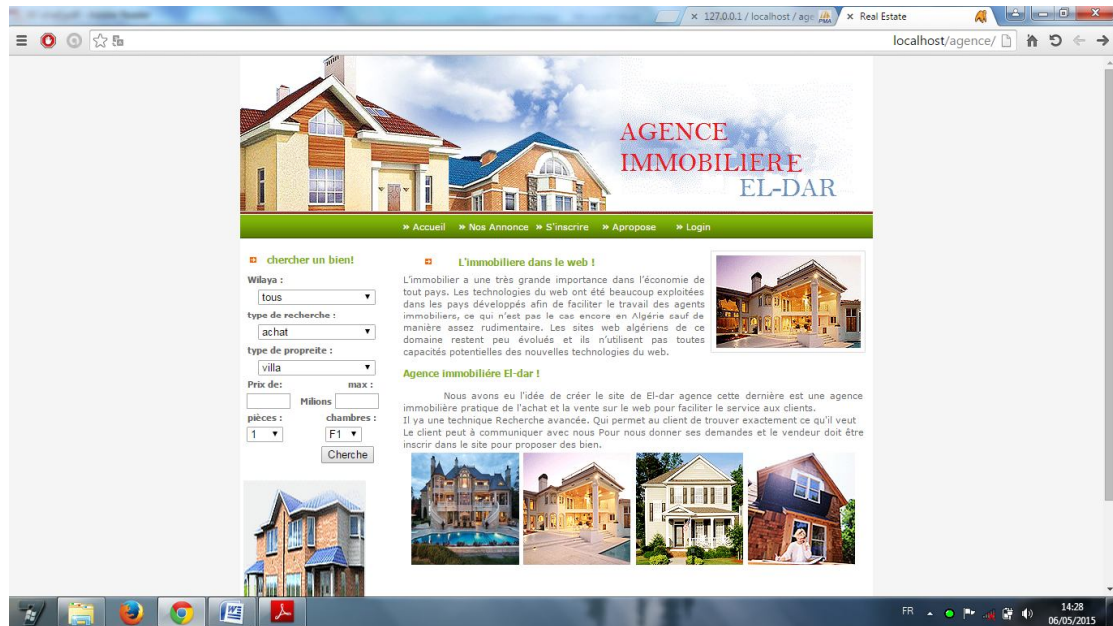


Figure 4.4 : page d'accueil

4.6.2 Nos annonces

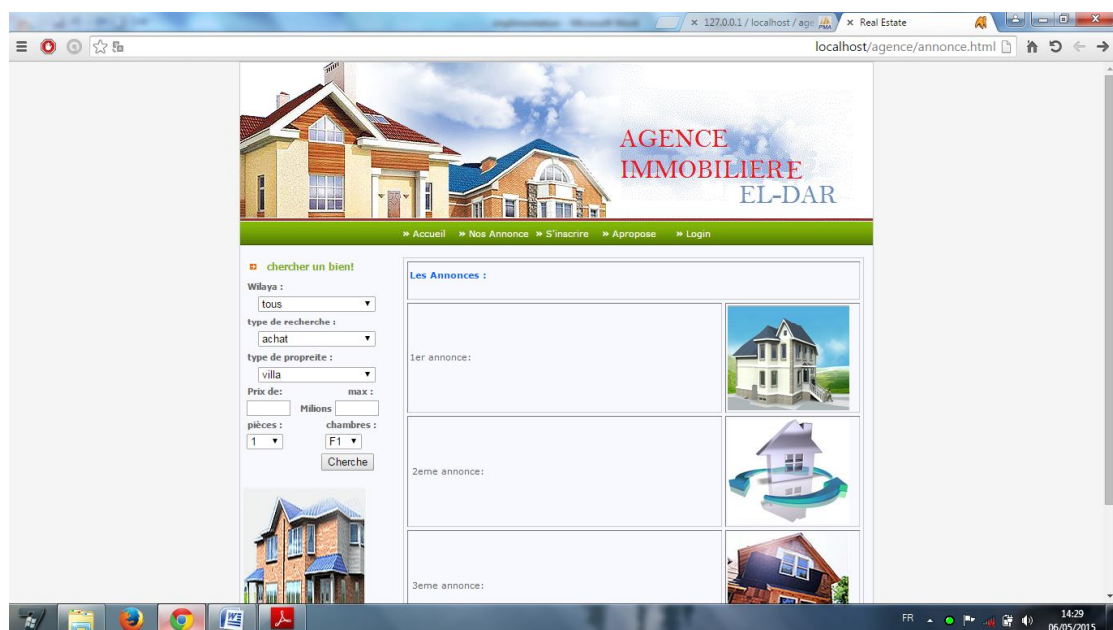


Figure 4.5 : nos annonces

4.6.3 Page d'inscrire :

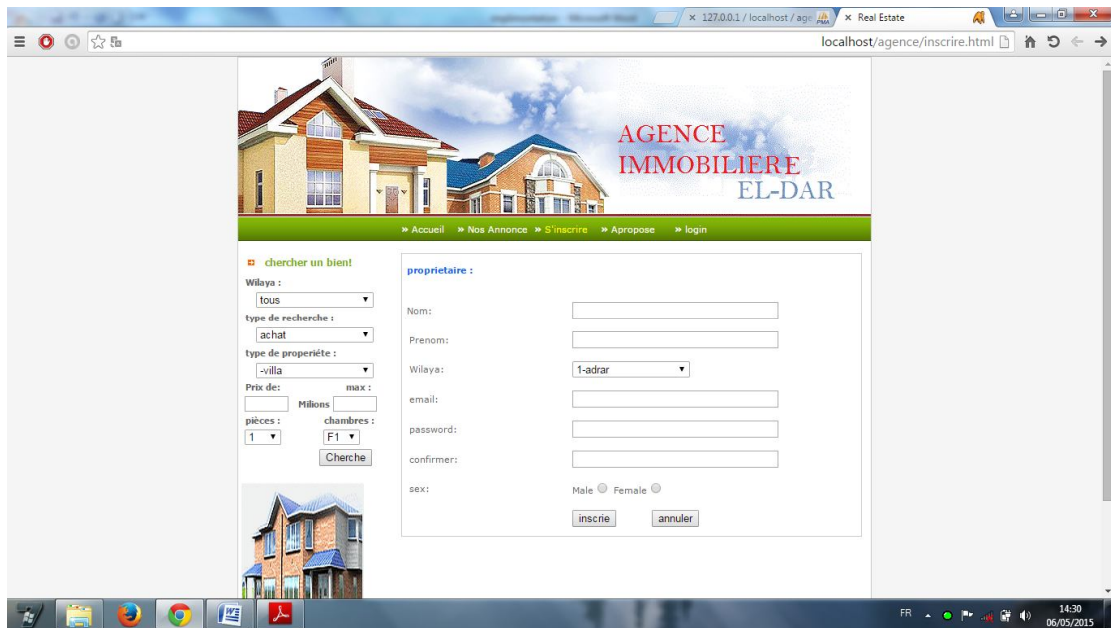


Figure 4.6 : page d'inscrire

4.6.4 A propose :

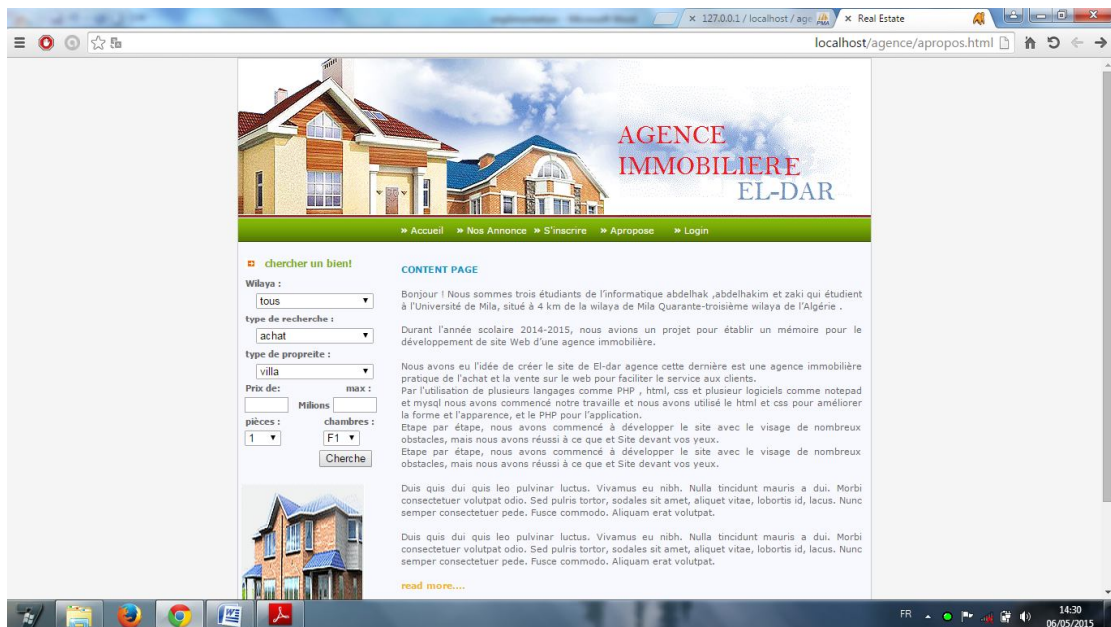


Figure 4.7 : A propose

4.6.5 Login :

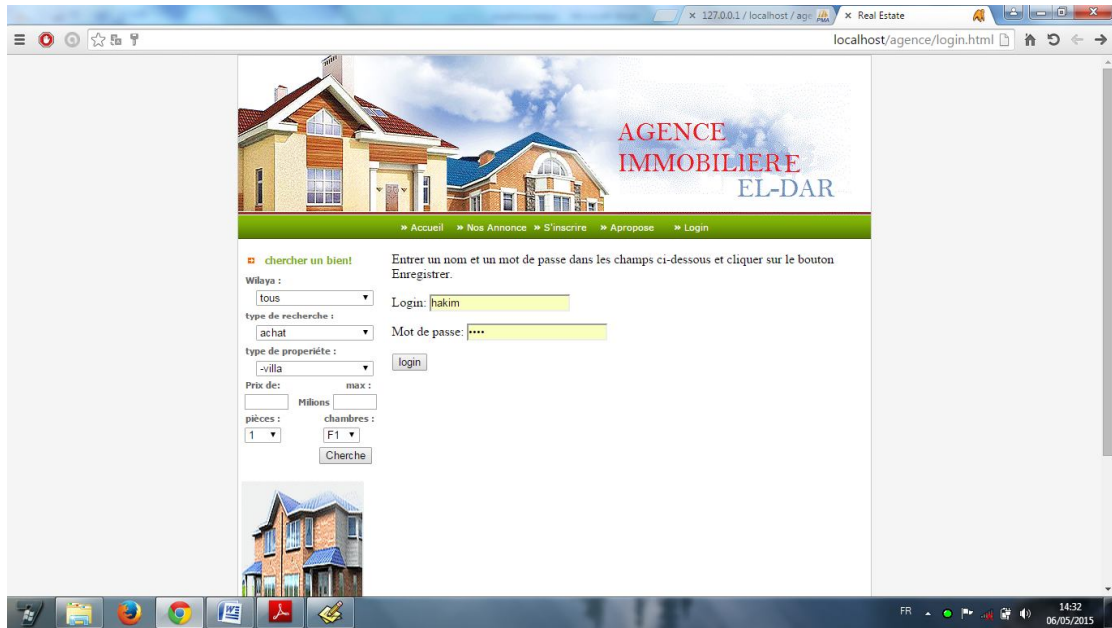


Figure 4.8 : log in

4.6.6 espace propriétaire :

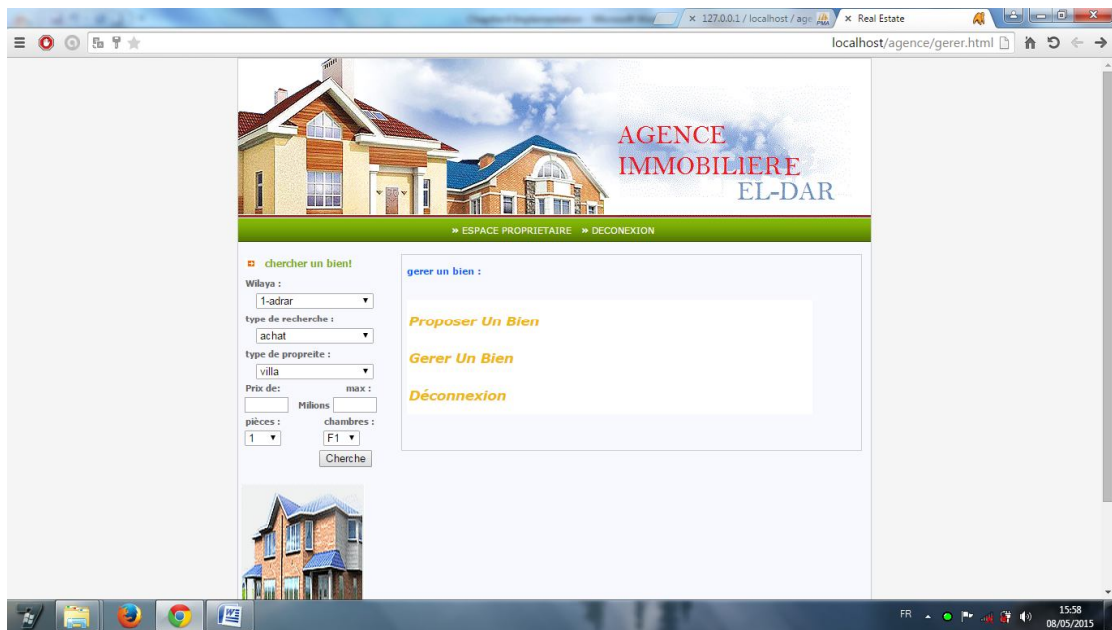


Figure 4.9 : Espace propriétaire

4.6.7 proposer un bien :

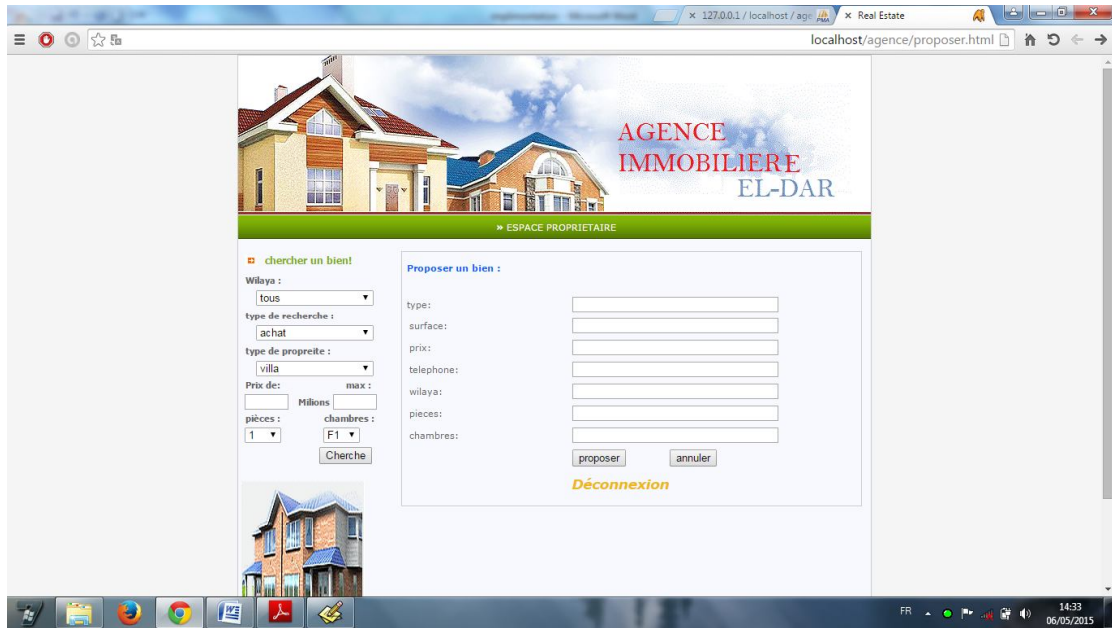


Figure 4.10 : proposer un bien

4.6.8 Gérer un bien

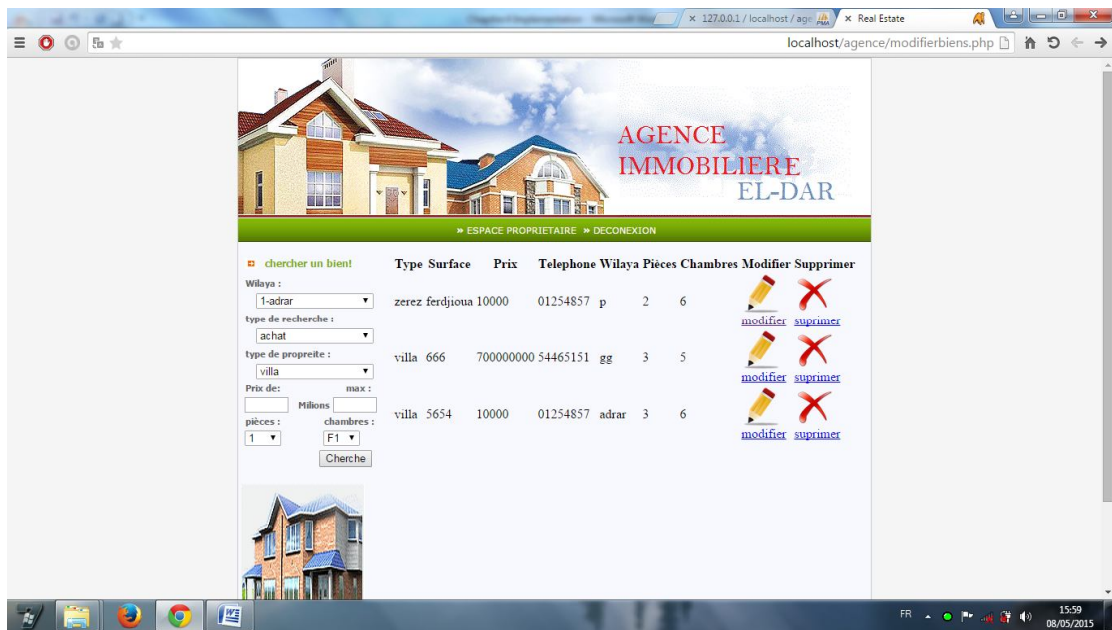


Figure 4.11 : Gérer un bien

4.6.9 supprimer un bien

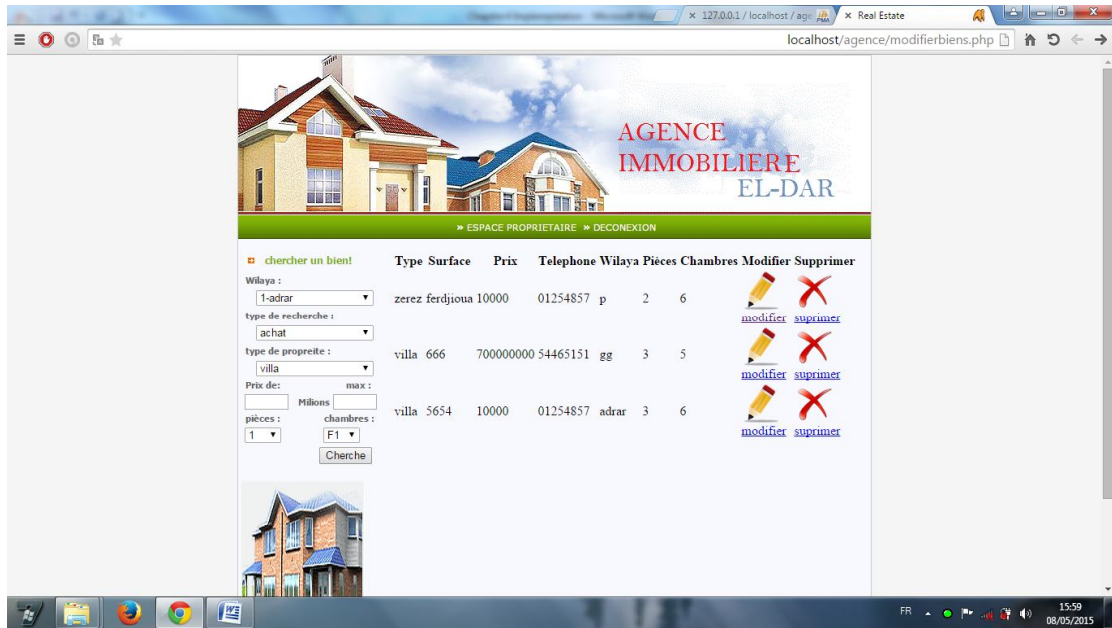


Figure 4.12 : Supprimer un bien

4.6.10 modifier un bien

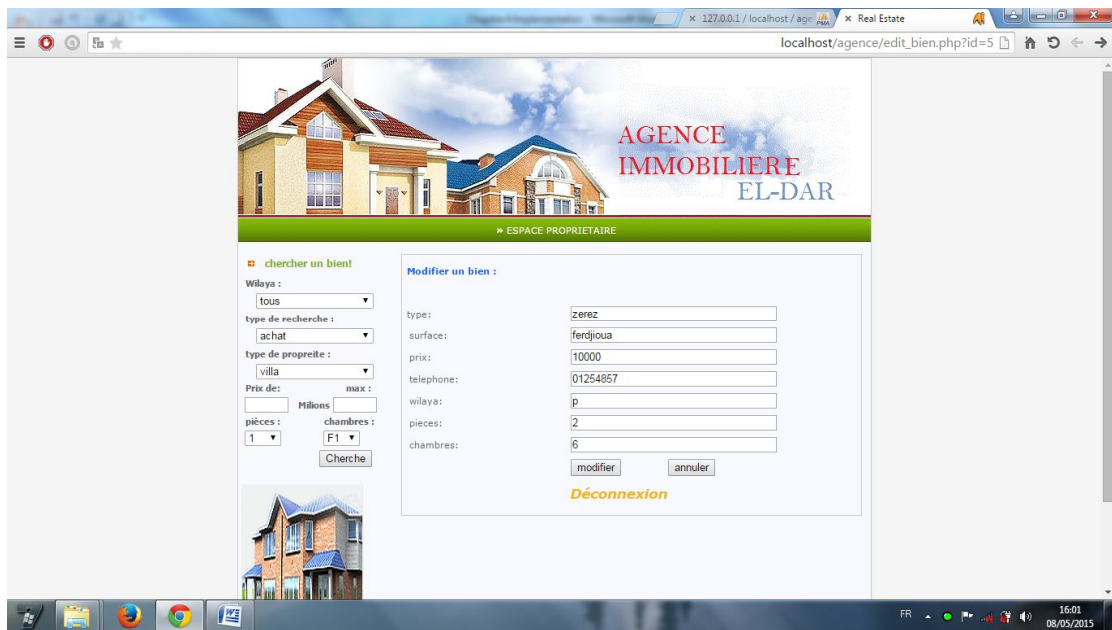


Figure 4.13 : modifier un bien

4.6.11 Choisir une annonce

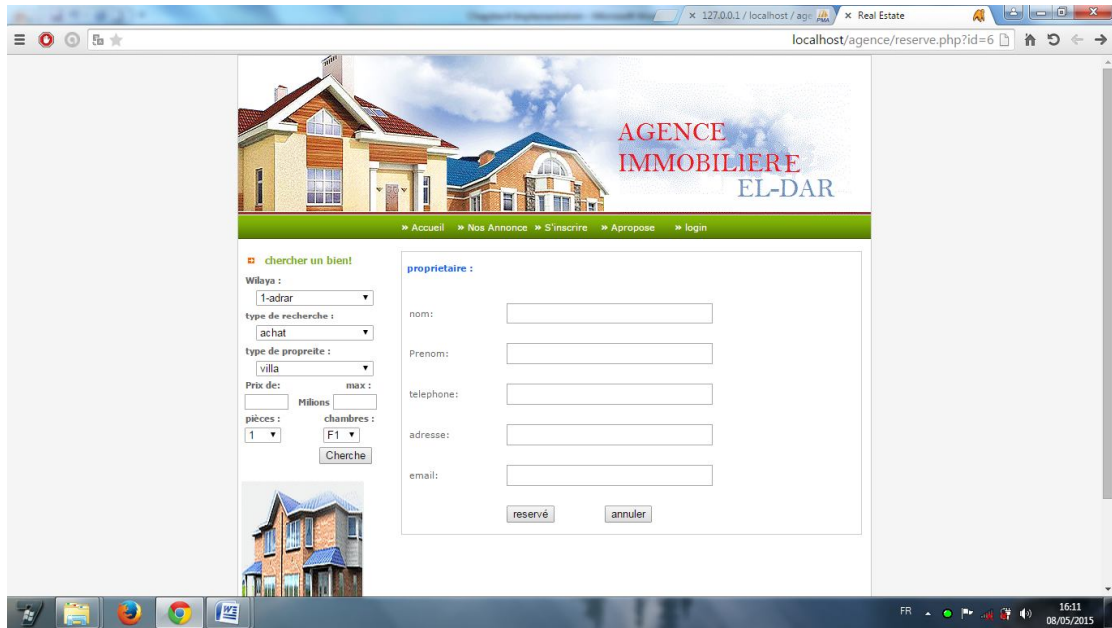


Figure 4.14 : choisir une annonce

4.6.12 gérer les comptes

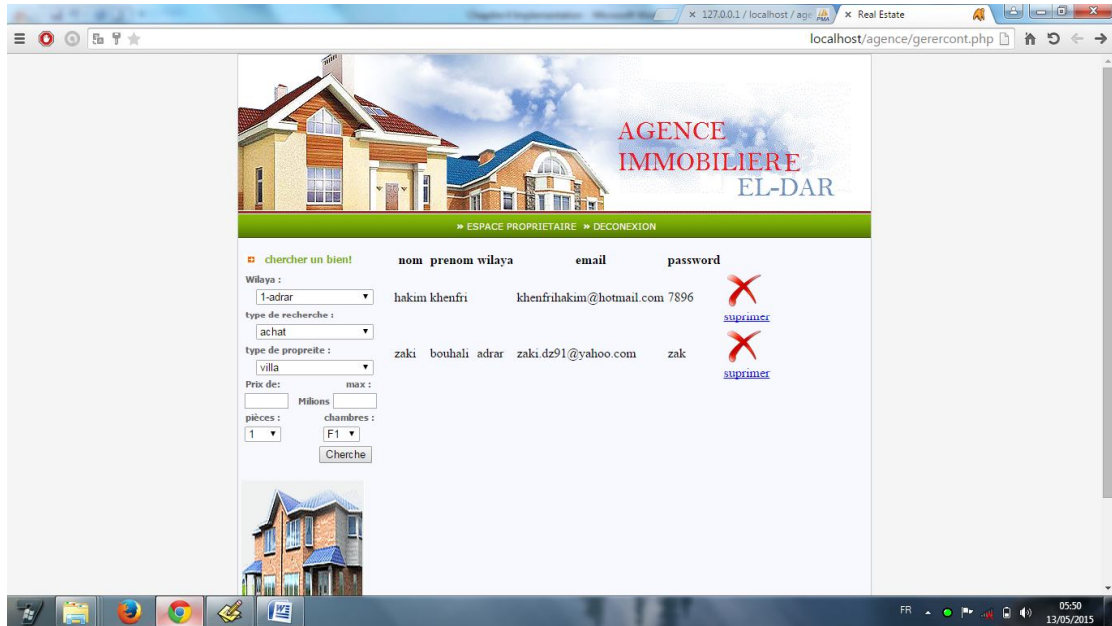


Figure 4.15 : Gérer les comptes

4.7 Conclusion

Ce chapitre est la phase finale dans notre travail. Il comprend une description générale des langages et des outils utilisés pour le développement du site. Par ailleurs, il comporte les interfaces qui déterminent de façon plus claire les activités réalisées dans ce site. Finalement, on peut dire que notre application répond à l'objectif souhaité dès le départ.

CONCLUSION GENERALE

Le travail présenté dans ce mémoire est la conception et la réalisation d'un site web d'une agence immobilière qui s'appelle **EL-DAR**.

Pour la réalisation de ce site, nous avons utilisé le langage UML pour la phase de conception. Pour la phase d'implémentation, nous avons utilisé le langage de programmation PHP, le serveur de base de données MySQL et le serveur APACHE, dans le trio connu sous le nom d'EasyPHP.

La réalisation de ce travail est une expérience enrichissante. D'une part, elle nous a permis d'acquérir des connaissances théoriques et pratiques en langages de programmation. Ainsi, nous avons pu mettre en pratique tous ce que nous avons appris au cours de l'année. D'autre part, elle nous a permis de nous affronter au monde professionnel.

Nous espérons que les objectifs fixés au départ ont été atteints et que le site répondra aux besoins des clients.

Finalement, nous pouvons dire que le développement d'une application qui soit, tout à la fois, contrôlée et renouvelable exige la mise en œuvre de connaissances pratiques éprouvées, telles que le développement itératif, la modélisation graphique et en particulier l'architecture en couches.

REFERENCES BIBLIOGRAPHIQUES

[1] www.wikipedia.com

[2] www.siteduzero.com

[3] www.fr.wikipedia.org

[4] www.uml-sysnl.org

[5] Laurent-peichoki.developpeur.com/tutoriel

[6] Uml.free.com

[7] Laurent AUDIBERT UML 2 Édition 2007-2008

[8] Grand livre HTML Daniel KOCH, Oliver KURTEN, Florian HARMS
1ère édition 2000

[9] A.Stallek, M.Kreinake, Java script-programmation

[10] Philippe Rigaux, Pratique de MySQL et PHP

[11] l'essentiel pour concevoir un site web dynamique