



REPUBLIQUE DU BENIN

MINISTRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

UNIVERSITE D'ABOMEY CALAVI (UAC)



ECOLE NATIONALE D'ECONOMIE APPLIQUÉE ET DE MANAGEMENT (ENEAM)

MÉMOIRE DE FIN DE FORMATION POUR L'OBTENTION DU DIPLÔME DE
TECHNICIEN SUPERIEUR (DTS)

Filière : Informatique de Gestion (IG)

Spécialité : Analyse Informatique Et
Programmation (AIP)

THEME

**Création d'une application de recherche et de
réservation de logements.**

Réalisé par : PADONOU Dieu-Donné et TOSSOU Cyriaque

Sous la direction de :

Maître de stage :

Luckmann GNAGOLI

Développeur backend à acumen

Fabrice KIKI

Développeur mobile à acumen

Maître de mémoire :

Dr. Maurice Comlan

Enseignant à l'ENEAM

ANNEE ACADEMIQUE : 2018 - 2019

DEDICACES

REMERCIEMENTS

Ce travail est le fruit des efforts conjugués de bienveillantes personnes à qui nous devons toute notre reconnaissance et gratitude. Nos remerciements vont particulièrement à l'endroit de :

- ❖ Mme Rosaline D. WOROU HOUNDEKON, Directrice de l'ENEAM ;
- ❖ M. Théophile K. DAGBA, PhD., Directeur-adjoint, Chargé des affaires académiques de l'ENEAM ;
- ❖ M. Maurice COMLAN, mon encadreur qui, malgré ses multiples occupations, a accepté de diriger ce travail et y a accordé une attention particulière ;
- ❖ M. Luckmann GNAGOLI et M. Fabrice KIKI, nos maitres de stage ;
- ❖ Nos enseignants, qui ont su nous inculquer le goût du travail bien fait ;
- ❖ Enfin, tous ceux qui, de près ou de loin ont contribué à la réalisation ce travail.

RESUME

ABSTRACT

Liste des tableaux

TABLEAU 1: IDENTIFICATION DES ACTEURS ET DES CAS D'UTILISATION DE HOHAYA.....	17
---	----

Liste des Figures

FIGURE 1: ORGANIGRAMME DE ACUMEN NETWORK.....	11
FIGURE 2: DIAGRAMME DE CAS D'UTILISATION DE HOHAYA.....	18
FIGURE 3: DIAGRAMME DE CLASSE DE HOHAYA.....	22
FIGURE 4: DIAGRAMME D'ETAT TRANSITION DU CAS D'UTILISATION "PUBLIER UN LOGEMENT"	25

LISTE DES SIGLES ET ABREVIATIONS

- ❖ **API** : Application Programming Interface.
- ❖ **BD** : Base de données.
- ❖ **ENEAM** : Ecole Nationale d'Economie Appliquée et de Management.
- ❖ **JSON** : JavaScript Object Notation.
- ❖ **JWT** : Json Web Token.
- ❖ **REST** : REpresentational State Transfer.
- ❖ **SGBD** : Système de Gestion de Base de Données.
- ❖ **SQL** : Structured Query Language.
- ❖ **UML** : Unified Modeling Language.

TABLE DES MATIERES

INTRODUCTION	9
Chapitre I : Présentation du contexte de l'étude	10
<i>1.1 Présentation de Acumen Network</i>	<i>11</i>
1.1.1 Historique et mission	11
1.1.2 Organigramme	11
1.1.3 Déroulement du stage.....	12
Chapitre II : Modélisation et spécification des besoins.....	13
<i>2.1 Spécification des besoins.....</i>	<i>14</i>
2.1.1 Présentation des acteurs	14
2.1.2 Spécifications fonctionnelles	14
2.1.3 Description fonctionnelle	14
2.1.4 Spécifications non-fonctionnelles.....	16
<i>2.2 Analyse orientée objet.....</i>	<i>16</i>
2.2.1 Le diagramme de cas d'utilisation.....	17
2.2.2 Classes et modèle relationnel.....	21
2.2.3 Diagrammes de séquence.....	24
2.2.4 Diagrammes d'état transition.....	24
2.2.5 Diagramme d'activité	25
Chapitre III : Implémentation du système	26
CONCLUSION	27
BIBLIOGRAPHIE & WEBOGRAPHIE	28

INTRODUCTION

Chapitre I : Présentation du contexte de l'étude

1.1 Présentation de Acumen Network

1.1.1 Historique et mission

Acumen Network est une entreprise franco-béninoise créée en 2016. Elle fournit à ses clients un soutien holistique en vue de les accompagner dans leurs ambitions de transformations technologiques et digitales.

De façon plus spécifique, l'entreprise met à la disposition de ses clients son expertise dans les domaines du business et de la mise en œuvre de projets IT. Elle propose également des formations et des séances de coachings.

1.1.2 Organigramme

Le fonctionnement d'Acumen network est organisé selon la hiérarchie décrite par l'organigramme ci-dessous.

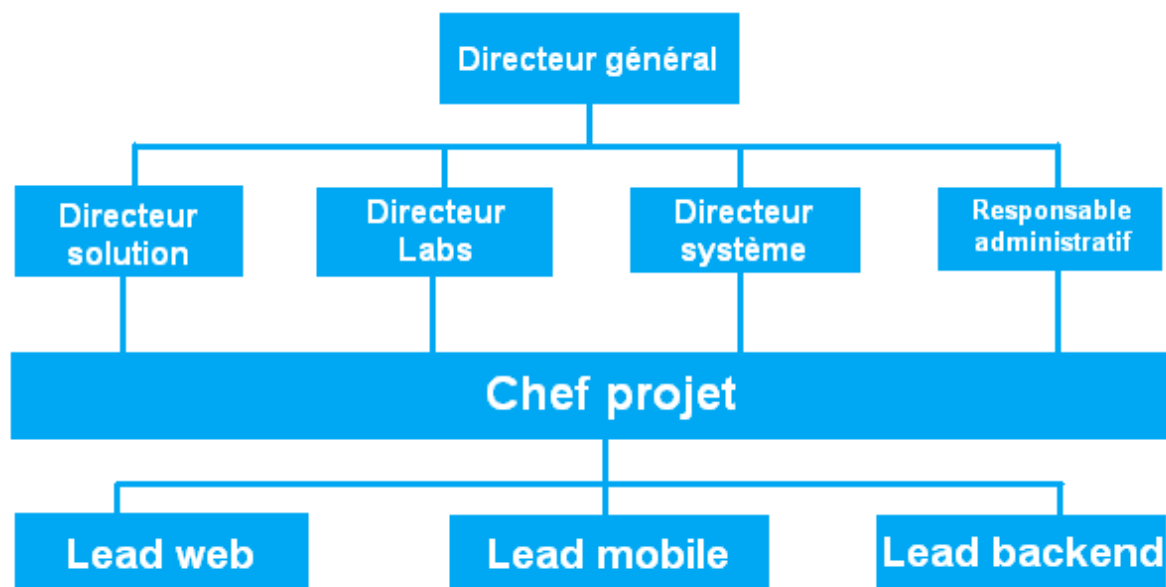


Figure 1: Organigramme de Acumen network

1.1.3 Déroulement du stage

Notre stage à Acumen a débuté le 27 octobre 2019 et a pris fin le 27 janvier 2020. Durant cette période, hormis le développement de notre application, nous avons eu l'opportunité de participer à certains travaux au sein de l'entreprise en réalisant les tâches qui nous étaient confiées. Nous avons aidé à peupler la base de données utilisée pour tester la plateforme digitale Agrosfer. Nous avons également été formé à l'utilisation de quelques technologies et outils tels que le Framework Java Spring pour la création des microservices et des application web, les technologies Java et Flutter pour le développement d'applications mobiles et iOS, le Framework frontend Angular pour le développement web et le logiciel de design (UI/UX design) Figma pour la réalisation des interfaces utilisateurs.

Chapitre II : Modélisation et spécification des besoins

2.1 Spécification des besoins

2.1.1 Présentation des acteurs

Un acteur est une entité qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé ^[1]. Les acteurs présents dans notre système sont les suivants :

- ❖ Le client
- ❖ Le propriétaire

2.1.2 Spécifications fonctionnelles

Le système à mettre en place devra fournir certaines fonctionnalités. Il doit permettre :

- ❖ Au client :
 - De passer en revue les publications disponibles.
 - De rechercher un type de logement spécifique.
 - D'émettre une annonce.
 - De contacter un propriétaire pour un logement donné.
 - De réserver un logement pour une période déterminée.
 - D'apprécier un logement (Noter ou ajouter des commentaires).
- ❖ Au propriétaire :
 - D'ajouter de nouveaux logements.
 - De publier ses logements.
 - D'apporter des modifications ou de supprimer un logement.
 - D'accepter une réservation.

2.1.3 Description fonctionnelle

Hohaya a pour but de faciliter les opérations de recherche ou de réservation de logements. Ainsi il retrace toutes les étapes nécessaires à la prise de contact avec un propriétaire ou à la

réservation effective d'un logement tout en occultant les tâches encombrantes et inutiles ou en simplifiant les tâches redondantes. Hohaya permettra donc :

❖ De créer des comptes utilisateur :

Cette fonctionnalité permet à tout utilisateur de créer un compte permettant de l'identifier dans le système et de lui attribuer des privilèges selon qu'il est propriétaire ou client.

❖ D'ajouter un logement :

Cette fonctionnalité permet aux propriétaires d'ajouter de nouveaux logements dans le système en fournissant des images et des caractéristiques propres à chaque nouveau logement.

❖ De publier un logement :

Cette fonctionnalité permet à un propriétaire rendre un de ses logement visible pour tous les utilisateurs du système quand les conditions de son abonnement le lui permettent.

❖ De faire une réservation :

Cette fonctionnalité permet à un utilisateur de réserver un logement de son choix pour une période donnée.

❖ De rechercher un logement :

Cette option permet à tout utilisateur d'effectuer une recherche pour trouver un logement présentant des caractéristiques très précises.

❖ De contacter un propriétaire :

Cette option permet à un client d'obtenir les contacts d'un propriétaire pour un logement particulier.

❖ De répondre à une réservation :

Cette fonctionnalité permet à un propriétaire de d'accepter ou non la réservation d'un client.

❖ De publier des annonces :

Cette fonctionnalité permet à un client de créer des annonces présentant les caractéristiques d'un logement susceptible de lui convenir et donc de recevoir des notifications dès qu'un logement correspondant à la plupart des critères énumérés est publié.

2.1.4 Spécifications non-fonctionnelles

En plus des fonctionnalités citées ci-dessus, l'application doit répondre aux critères suivants, qui garantiront sa fiabilité et son utilisabilité :

LA SECURITE ET L'INTEGRITE DES DONNES : En raison du caractère hautement sensible des données à traiter, l'accès à certaines fonctionnalités de l'application n'est permis qu'après un processus d'authentification visant à s'assurer de l'identité et des droits de l'utilisateur en cause. Les informations d'inscription et d'authentification doivent être confidentielles. L'application doit garantir l'intégrité, la cohérence et la persistance des données.

LA RAPIDITE DE TRAITEMENT : Compte tenu de nombre élevé de traitements à effectuer à chaque instant il est impératif de veiller à réduire le délai d'exécution des différentes requêtes au minimum.

LA PERFORMANCE : Notre application doit être avant tout performant, c'est-à-dire à travers ses fonctionnalités, répondre à toutes les exigences des usagers d'une manière optimale.

LA CONVIVIALITÉ : Hohaya doit proposer des options faciles d'accès et d'utilisation. En effet, les interfaces utilisateurs doivent respecter les différentes normes ergonomiques en la matière et en matière d'expérience utilisateur.

2.2 Analyse orientée objet

Dans le cadre de l'analyse orientée objet du système nous avons opter pour le langage de modélisation unifié, plus connu sous la désignation United Modeling Language (UML) en anglais.

UML 2.3 propose 14 diagrammes subdivisés en 2 grandes catégories : les diagrammes dynamiques ou comportementaux et les diagrammes statique ou structurels. Dans ce travail d'analyse nous ne présenterons que 5 d'entre ces diagrammes. Il s'agit des diagrammes de cas d'utilisation, de classe, de séquence, d'état transition et d'activité.

2.2.1 Le diagramme de cas d'utilisation

2.2.2.1 Identification des cas d'utilisation de Hohaya.

Il est nécessaire de définir, pour chacun des acteurs, les actions à mener (cas d'utilisation). Les divers acteurs intervenant dans le système ainsi que les actions qu'ils peuvent mener sont représentés dans le tableau ci-dessous.

Tableau 1: Identification des acteurs et des cas d'utilisation de Hohaya

Acteurs	Cas d'utilisation
Client	<ul style="list-style-type: none">- S'inscrire- Se connecter- Gérer son compte- Apprécier une publication- Rechercher un logement- Gérer ses annonces- Ajouter une réservation- Contacter un propriétaire
Propriétaire	<ul style="list-style-type: none">- S'inscrire- Se connecter- Gérer son compte- Apprécier une publication- Rechercher un logement- Gérer ses annonces- Ajouter une réservation

	<ul style="list-style-type: none"> - Contacter un propriétaire - Gérer ses logements - S'abonner - Répondre à une réservation
--	---

2.2.2.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet d'exprimer les besoins des utilisateurs du système.

La figure suivante illustre le diagramme de cas d'utilisation de Hohaya.

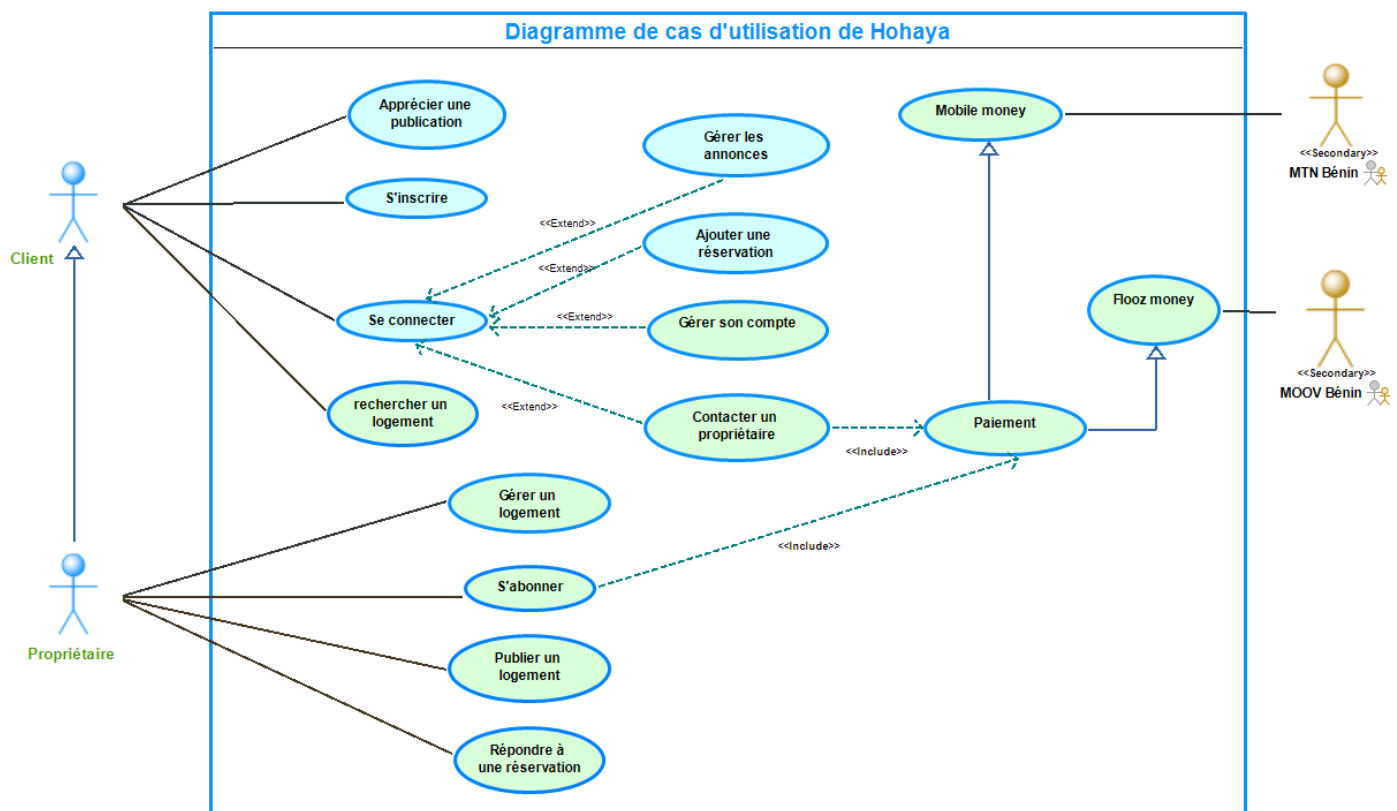


Figure 2: Diagramme de cas d'utilisation de Hohaya

Suite à une analyse approfondie du système nous avons élaborer le diagramme de cas d'utilisation ci-dessus qui résume aussi bien les interactions des différents acteurs avec le système que les interactions entre différents cas d'utilisation.

2.2.2.3 Description textuelle des cas d'utilisation

La description textuelle des cas d'utilisation nous permet d'établir de manière précise toutes les interactions qui auront lieu entre les acteurs et le système au cours de la réalisation d'un cas d'utilisation particulier.

Description textuelle du cas d'utilisation « **Contacter un propriétaire** »

Sommaire d'identification

Titre : Contacter un propriétaire

Résumé : Ce cas permet aux utilisateurs d'obtenir les contacts du propriétaire d'un logement donné

Acteurs : Client ou Propriétaire

Date de Création : 25/01/2020

Version : 1.0

Responsable : PADONOU Dieu-Donné et TOSSOU Cyriaque

Préconditions : Aucune

Description des Scénarii

▪ **Scénario nominal**

1. L'acteur clique sur le bouton "Contacter le propriétaire".
2. Le système vérifie que l'utilisateur est connecté.
3. Le système affiche le formulaire de paiement.
4. L'acteur renseigne les informations de paiement.
5. Le système vérifie la validité des informations renseignées.
6. Le système retire le montant prévu du compte de l'utilisateur.
7. Le système affiche les contacts du propriétaire.

▪ **Scénario alternatif**

A1 : Un (ou plusieurs) champs non renseigné(s) ou invalide(s)

Le scénario A1 démarre au point 5 du scénario nominal.

Le point 6 devient « Le système indique à l'acteur que certains champs sont requis ou invalides ».

Le scénario reprend au point 3 du scénario nominal.

A2 : L'utilisateur n'est pas connecté.

Le scénario A2 démarre au point 2 du scénario nominal.

Le point 3 devient « le système affiche la page de connexion ».

Le scénario reprend au point 3 du scénario nominal.

Description textuelle du cas d'utilisation « **Publier un logement** »

Sommaire d'identification

Titre : Publier un logement

Résumé : Ce cas permet aux propriétaires de publier un logement déjà existant dans le système.

Acteurs : Propriétaire

Date de Création : 25/01/2020

Version : 1.0

Responsable : PADONOU Dieu-Donné et TOSSOU Cyriaque.

Préconditions : Acteur authentifié comme Propriétaire.

Description des Scénarii

Scénario nominal

1. Le propriétaire clique sur le bouton 'Publier'.
2. Le système vérifie que son abonnement est actif.
3. Le système vérifie que les conditions de son abonnement lui permettent de publier ce logement.
4. Le système crée une nouvelle instance de publication pour ce logement dans la base de données.
5. Le système affiche un message de succès.

Scénario alternatif

A1 : Le propriétaire n'a aucun abonnement actif.

Le scénario A1 démarre au point 2 du scénario nominal.

Le point 3 devient « Le système indique au propriétaire qu'il n'a aucun abonnement actif ».

Le point 4 devient « Le système déclenche l'exécution du cas d'utilisation : S'abonner ».

Le scénario reprend au point 4 du scénario nominal.

Création d'une application de recherche et de réservation de logements.

A2 : Les conditions de son abonnement ne lui permettent pas de publier ce logement.

Le scénario A2 démarre au point 3 du scénario nominal.

Le point 4 devient « Le système lui notifie que son abonnement ne prend pas en charge cette publication ».

Le cas d'utilisation prend fin.

Postconditions :

Le statut de la publication du logement passe de 'inactif' à 'actif'.

Le logement est désormais visible pour tous les utilisateurs du système.

2.2.2 Classes et modèle relationnel

2.2.2.1 Diagramme de classes

Le diagramme de classe décrit clairement la structure du système en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. Les classes issues de l'analyse du système sont regroupés dans le diagramme ci-dessous avec les relations qui les lient.

Diagramme de classe de Hohaya

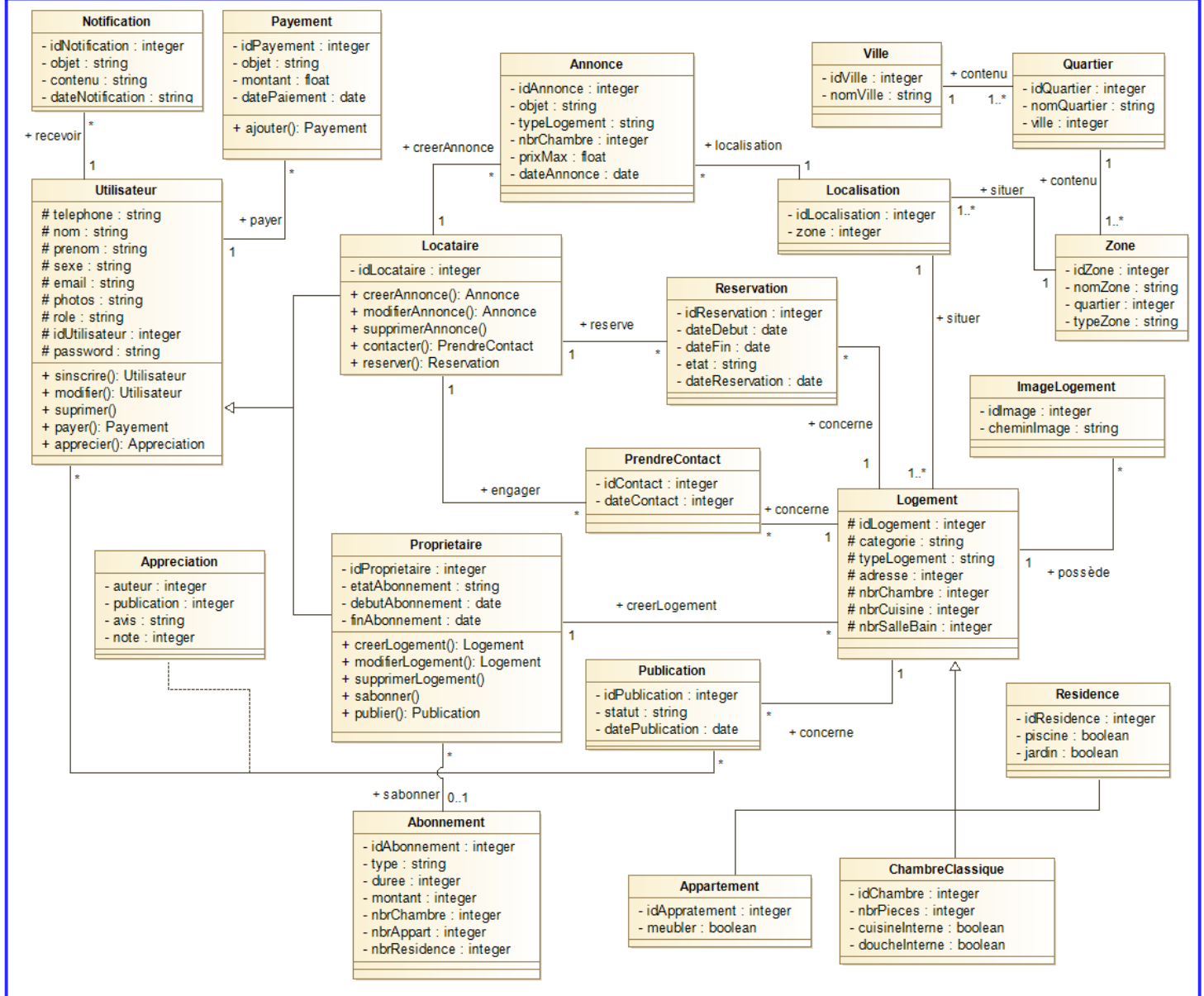


Figure 3: Diagramme de classe de Hohaya

2.2.2.2 Modèle relationnel du système

Le modèle relationnel est une manière de modéliser les informations, et de les ordonner entre elles. Il permet d'avoir un aperçu des différentes classes qui constitueront la base de données

du système et des attributs qui composeront chaque table. Le modèle relationnel en 3FN de notre système est le suivant :

- Locataire (idLocataire, nom, prenom, telephone, password, sexe, email, photo, role).
- Proprietaire (idProprietaire, nom, prenom, telephone, password, sexe, email, photo, role, etatAbonnement, debutAbonnement, finAbonnement, #idAbonnement).
- Abonnement (idAbonnement, type, duree, montant, nbrChambre, nbrResidence, nbrAppart).
- Notification (idNotification, objet, contenu, dateNotification, # idUtilisateur).
- Paiement (idPaiement, objet, montant, datePaiement, # idUtilisateur).
- Annonce (idAnnonce, objet, typeLogement, nbrChambre, prixMax, dateAnnonce, # idUtilisateur, #idLocalisation).
- Reservation (idReservation, dateDebut, dateFin, etat, dateReservation, #idUtilisateur, #idLogement).
- Logement (idLogement, categorie, typeLogement, nbrChambre, nbrCuisine, nbrSalleBain, #idProprietaire, #idLocalisation).
- ChambreClassique (idChambre, nbrChambre, nbrPiece, cuisineInterne, doucheInterne).
- Residence (idResidence, jardin, piscine).
- Appartement (idAppart, meubler).
- Appreciation (#idUtilisateur, #idPublication, avis, note).
- Publication (idPublication, statut, datePublication, idLogement).
- ImageLogement (idImage, cheminImage, #idLogement).
- PrendreContact (idContact, dateContact, #idUtilisateur, #idLogement).
- Localisation (idlocalisation, #idZone).
- Zone (idZone, nomZone, typeZone, #idQuartier).
- Quartier (idQuartier, nomQuartier, #idVille).
- Ville (idVille, nomVille).

2.2.2.3 Règles de gestion

Les règles de gestion ci-dessous régissent le fonctionnement du système.

R1- Seul un propriétaire peut publier, modifier ou supprimer un logement.

- R2- Un propriétaire peut faire tout ce que peut faire un client.
- R3- Une appréciation est soit une note ou une note suivie d'un commentaire.
- R4- Un utilisateur ne peut émettre qu'une seule appréciation pour un logement donné.
- R5- Plusieurs logements peuvent avoir la même localisation.
- R6- Un même locataire peut prendre contact avec un propriétaire un le même logement plusieurs fois.
- R7- Un logement n'a qu'une publication qui lui est relative.
- R8- Un propriétaire peut ne pas souscrire à un abonnement.
- R9- Seuls les propriétaires peuvent souscrire à un abonnement.
- R10- Un propriétaire ne peut souscrire qu'à un seul abonnement à la fois.
- R11- Un logement n'est plus disponible pour la réservation dès qu'un propriétaire accepte une réservation faite pour ce logement.

2.2.3 Diagrammes de séquence

Un diagramme de séquence est diagramme d'interaction. Il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Les diagrammes ci-dessous donnent une représentation graphique des interactions décrites plus haut entre utilisateurs et système.

2.2.4 Diagrammes d'état transition

Un diagramme d'état transition est diagramme comportemental qui représente les transitions entre divers objets. Les états ici correspondent aux différentes combinaisons d'informations qu'un objet peut contenir. Ce diagramme présente un automate à état finis. Il permet ainsi donc de décrire les changements d'état d'un objet ou d'un composant. Les diagrammes d'état transition ci-dessous expliquent précisément de quelle façon les actions sont exécutées lors des transitions entre les différents états correspondant aux différentes situations se produisant lors de la réalisation de chaque cas d'utilisation.

Voici illustré ci-dessous quelques diagrammes d'état transition de Hohaya

- Diagramme d'état transition du cas « Publier un logement »

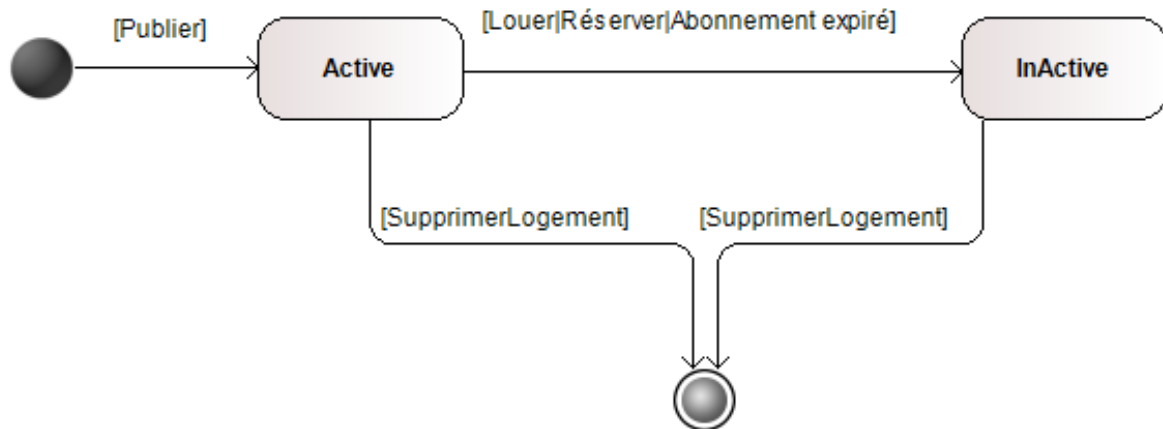


Figure 4: Diagramme d'état transition du cas d'utilisation "Publier un logement"

2.2.5 Diagramme d'activité

Chapitre III : Implémentation du système

CONCLUSION

BIBLIOGRAPHIE & WEBOGRAPHIE

[1] : [https://fr.wikipedia.org/wiki/Acteur_\(UML\)](https://fr.wikipedia.org/wiki/Acteur_(UML)) – consulté le 17 février 2020 à 15h11mn