REPUBLIQUE DU BENIN

MINISTERE 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 toute notre 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 19](#_Toc34027291)

Liste des Figures

[Figure 1: Organigramme de Acumen network 11](file:///D:\Workspace\Memoire\Memoire.docx#_Toc34027284)

[Figure 2 : Diagramme de cas d'utilisation de Hohaya 20](file:///D:\Workspace\Memoire\Memoire.docx#_Toc34027285)

[Figure 3 : Diagramme de classe de Hohaya 24](file:///D:\Workspace\Memoire\Memoire.docx#_Toc34027286)

[Figure 4: Diagramme de séquence du cas "Publier un logement" 27](file:///D:\Workspace\Memoire\Memoire.docx#_Toc34027287)

[Figure 5: Diagramme de séquence du cas "contacter un propriétaire" 29](file:///D:\Workspace\Memoire\Memoire.docx#_Toc34027288)

[Figure 6 : Diagramme d'état transition du cas d'utilisation « Publier un logement » 30](#_Toc34027289)

[Figure 7: Diagramme d'état transition d'une réservation 31](file:///D:\Workspace\Memoire\Memoire.docx#_Toc34027290)

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](#_Toc34027258)

[Chapitre I : Présentation du contexte de l’étude 10](#_Toc34027259)

[1.1 Présentation de Acumen Network 11](#_Toc34027260)

[1.1.1 Historique et mission 11](#_Toc34027261)

[1.1.2 Organigramme 11](#_Toc34027262)

[1.1.3 Déroulement du stage 12](#_Toc34027263)

[1.2 Etude préliminaire 12](#_Toc34027264)

[1.2.1 Présentation de l’existant 12](#_Toc34027265)

[1.2.2 Critique de l’existant 13](#_Toc34027266)

[1.2.3 Approche de solution : une application de recherche et de réservation de logement 14](#_Toc34027267)

[1.2.4 Objectifs globaux 14](#_Toc34027268)

[Chapitre II : Modélisation et spécification des besoins 15](#_Toc34027269)

[2.1 Spécification des besoins 16](#_Toc34027270)

[2.1.1 Présentation des acteurs 16](#_Toc34027271)

[2.1.2 Spécifications fonctionnelles 16](#_Toc34027272)

[2.1.3 Description fonctionnelle 16](#_Toc34027273)

[2.1.4 Spécifications non fonctionnelles 18](#_Toc34027274)

[2.2 Analyse orientée objet 18](#_Toc34027275)

[2.2.1 Le diagramme de cas d’utilisation 19](#_Toc34027276)

[2.2.2 Classes et modèle relationnel 23](#_Toc34027277)

[2.2.3 Diagrammes de séquence 26](#_Toc34027278)

[2.2.4 Diagrammes d’état transition 30](#_Toc34027279)

[2.2.5 Diagramme d’activité 31](#_Toc34027280)

[Chapitre III : Implémentation du système 32](#_Toc34027281)

[CONCLUSION 33](#_Toc34027282)

[BIBLIOGRAPHIE & WEBOGRAPHIE 34](#_Toc34027283)

# ****INTRODUCTION****

# Présentation du contexte de l’étude

## Présentation de Acumen Network

### Historique et mission

Acumen Network est une entreprise franco-béninoise crée 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.

### Organigramme

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

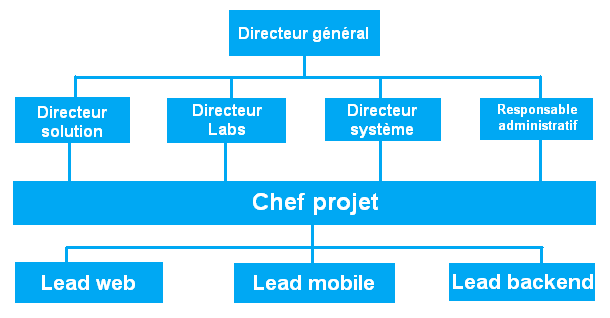


Figure 1: Organigramme de Acumen network

### 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.

## Etude préliminaire

### Présentation de l’existant

Les opérations de recherche, de location et de réservation de logement sont de plus en plus récurrentes. Sollicités par divers clients (Apprenants, Artisans, Fonctionnaires, toutes personnes en besoin de logement étranger ou non), les services de logements se font de maintes manières.

D’abord, de bouche à oreille et passant de maison en maison, certains locataires parviennent à identifier les logements disponibles pour location ou réservation.

En l’état actuel des choses, les clients en recherche de logement doivent nécessairement se rapprocher des agences immobilières de la place ou s’offrir les services ‘’démarcheurs’’, qui officient comme des agents immobiliers indépendants. Ces démarcheurs disposent au préalable des contacts de propriétaires de logement mis en location selon leurs types et les zones. Pour rendre ce service, les démarcheurs perçoivent des frais de déplacement pour chaque nouveau logement visité indépendamment de la satisfaction du client et un mois de loyer du logement lorsque celui-ci a été accepté par le client.

Outre cette manière de procéder, les clients internationaux peuvent se tournés vers les plateformes d’e-commerce au monde dans le secteur du voyage. Ces plateformes, à l’instar de Booking, Abritel, Expedia et biens d’autres encore, proposent, entre autres services, des logements dans plus d’une cinquantaine de pays.

### Critique de l’existant

La procédure de bouche à oreille et de passage de maison enmaison est très peu efficace et parfois hasardeuse. Elle nécessite énormément de travail avec l’incertitude de parvenir à un résultat concluant obligeant le client à avoir une parfaite maitrise des zones.

Recourir aux services d’un démarcheur peut se révéler très fastidieux et onéreux pour le client en besoin de logement, car il se trouve contraint d’assurer le déplacement du démarcheur pour tous les logements qu’ils auront à visiter faisant le tour de maison en maison jusqu’à trouver un logement qui lui convient. Ensuite, une fois le logement approprié trouvé, s’ils en trouvent, le client doit verser une somme équivalant à un mois de loyer du logement choisi, cette somme étant considérée comme frais du service rendu, il est indépendant des frais de location à verser au propriétaire.

Les services offerts par les plateformes de booking internationaux sont axés sur les voyages à travers le monde. En raison de leur clientèle composée majoritairement de particuliers devant visiter ou séjourner dans un autre pays, les logements proposés sont principalement les hôtels et les résidences de haut standing. Elles ne gèrent donc pas les locations de petite taille à l’intérieur d’un pays. De plus les moyens de paiement dont dispose les clients ne se limitent qu’à ceux employés de façon classique dans les transactions monétaires en ligne et ne prennent donc pas en compte les moyens moins connus mais bien plus utilisés au Bénin.

### Approche de solution : une application de recherche et de réservation de logement

Dans le but de simplifier les opérations de recherche et de réservation de logements nous proposons une solution qui rassemble les acteurs impliqués dans ce processus. Il s’agit d’une application mobile nommée Hohaya, qui mets à la disposition du propriétaire un espace lui permettant d’aller à la rencontre de ses locataires en publiant ses logements après avoir souscrit à un abonnement dérisoire. Hohaya permet au client de retrouver un logement correspondant exactement à ces attentes au bout d’une recherche de quelques minutes tout au plus. Il permet également au client de laisser des annonces afin de recevoir des notifications dès qu’un logement présentant les caractéristiques énumérées dans l’annonce est disponible. De plus, contrairement aux services existants dans ce domaine sur internet, les publications couvrent un large panel de logements allant des maisons dans les zones les plus reculées aux hôtels les plus connus et les moyens de paiement mis à la disposition de l’utilisateur sont les plus répandus sur dans le pays en matière de transaction financière en ligne, en l’occurrence Mobile money et Flooz.

### Objectifs globaux

Ce projet vise principalement à réduire, le plus possible, le temps et l’argent engager dans la recherche d’un logement par les particuliers. L’aboutissement de ce projet permettra donc, d’une part, au client de réaliser un gain de temps très conséquent lors des opérations de recherche d’un logement, et d’autre part, au propriétaire d’élargir sa clientèle cible en vue de trouver plus vite des occupants pour ses logements.

# Modélisation et spécification des besoins

## Spécification des besoins

### 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

### 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.

### 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 de rendre un de ses logements 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 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.

### 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 performante, 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.

## Analyse orientée objet

Dans le cadre de l’analyse orientée objet du système, nous avons opté 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 statiques 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.

### Le diagramme de cas d’utilisation

#### 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 | * S’inscrire * Se connecter * Gérer son compte * Apprécier une publication * Rechercher un logement * Gérer ses annonces * Ajouter une réservation * Contacter un propriétaire |
| Propriétaire | * S’inscrire * Se connecter * Gérer son compte * Apprécier une publication * Rechercher un logement * Gérer ses annonces * Ajouter une réservation * Contacter un propriétaire * Gérer ses logements * S’abonner * Répondre à une réservation |

#### 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.

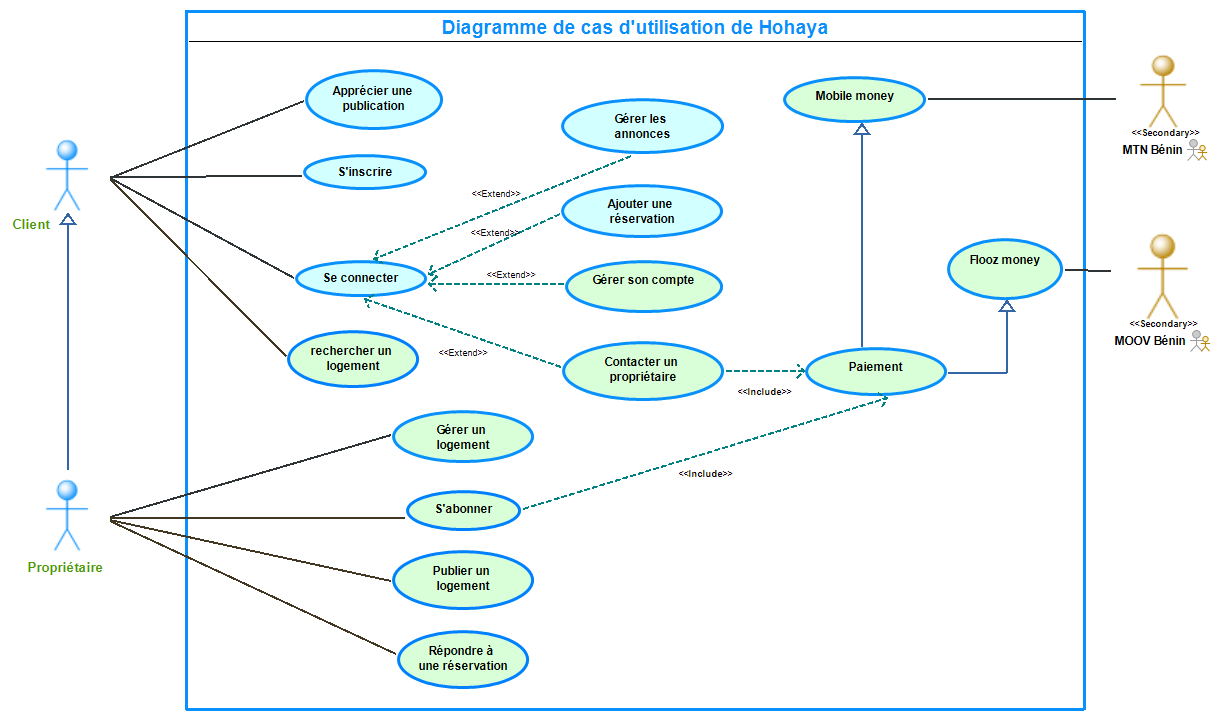


Figure 2 : Diagramme de cas d'utilisation de Hohaya

Suite à une analyse approfondie du système, nous avons élaboré 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.

#### 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) champ 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.

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.

### Classes et modèle relationnel

#### 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.

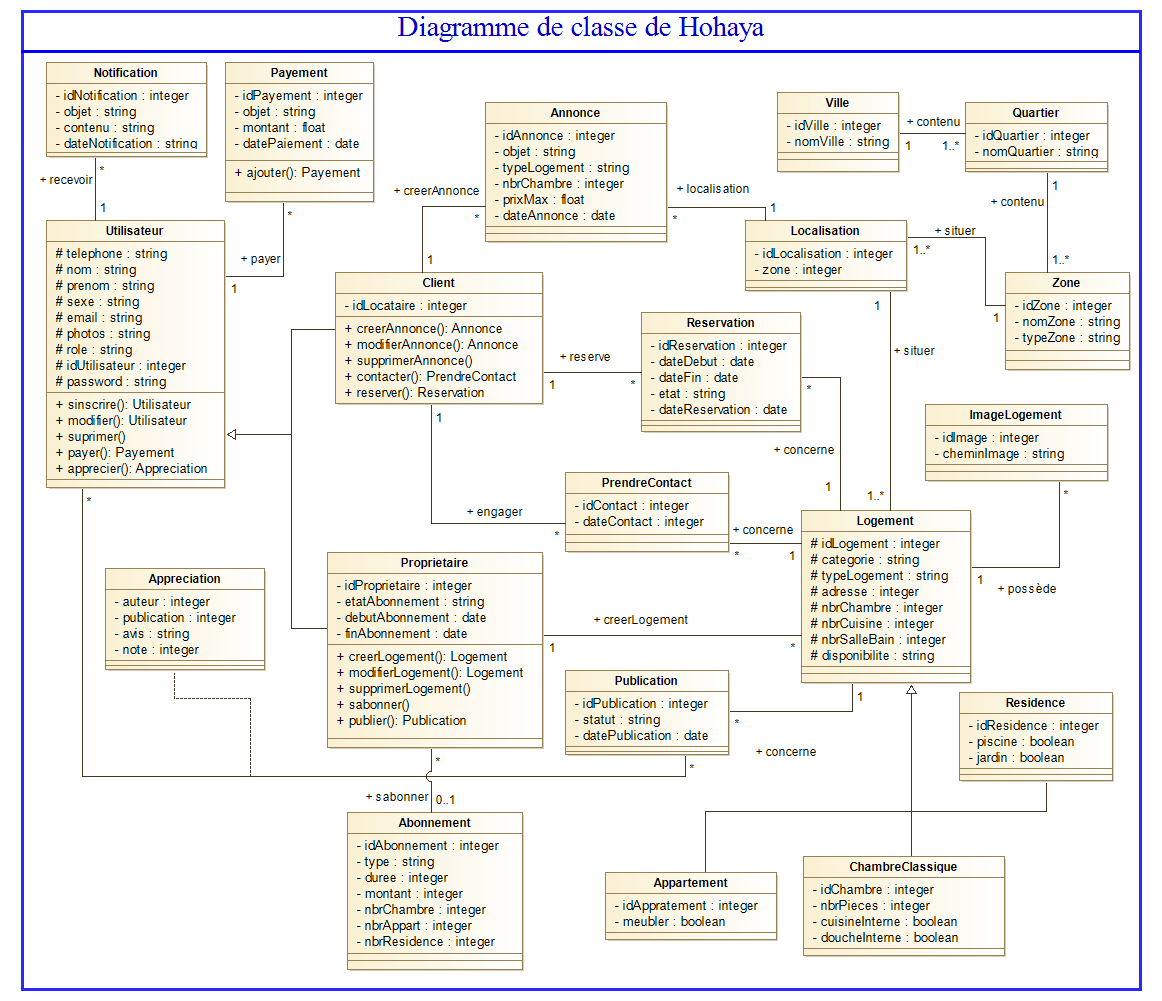


Figure 3 : Diagramme de classe de Hohaya

#### 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 (idLoacataire, 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).

#### Règles de gestion

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

1. Seul un propriétaire peut publier, modifier ou supprimer un logement.
2. Un propriétaire peut faire tout ce que peut faire un client.
3. Une appréciation est soit une note ou une note suivie d’un commentaire.
4. Un utilisateur ne peut émettre qu’une seule appréciation pour un logement donné.
5. Plusieurs logements peuvent avoir la même localisation.
6. Un même locataire peut prendre contact avec un propriétaire pour le même logement plusieurs fois.
7. Un logement n’a qu’une publication qui lui est relative.
8. Un propriétaire peut ne pas souscrire à un abonnement.
9. Seuls les propriétaires peuvent souscrire à un abonnement.
10. Un propriétaire ne peut souscrire qu’à un seul abonnement à la fois.
11. Un logement n’est plus disponible pour la réservation dès qu’un propriétaire accepte une réservation faite pour ce logement.

### 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.

* Diagramme de séquence du cas « **publier un logement** ».

Le diagramme ci-dessous illustre le processus aboutissant à la publication d’un logement. Ainsi après avoir cliqué sur le bouton « **Publier** », le système vérifie que le propriétaire dispose d’un abonnement en cours de validité au quel cas le système vérifie si l’abonnement du propriétaire lui permet de faire cette publication. Si oui, alors pour une première publication, une nouvelle instance de publication est créée en base pour ce logement et l’état de cette publication est mis à « **actif** ». Si l’abonnement du propriétaire ne couvre pas cette publication alors une notification est envoyée par le système pour en avertir le propriétaire et la publication n’est pas faite. Dans le cas où le propriétaire n’a pas du tout d’abonnement il est redirigé vers l’interface de souscription à un nouvel abonnement.

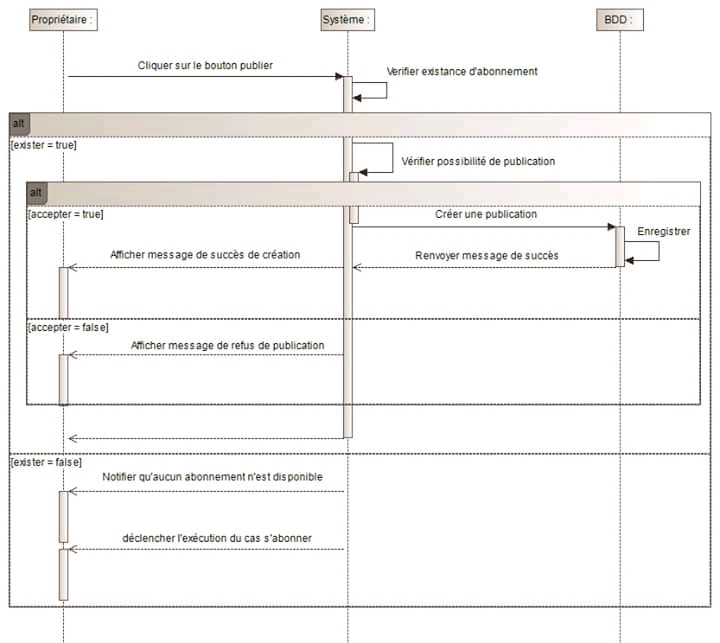


Figure 4: Diagramme de séquence du cas "Publier un logement"

* Diagramme de séquence du cas « **contacter un propriétaire** »

Le diagramme suivant illustre le fonctionnement du cas d’utilisation permettant à un client de contacter un propriétaire. Après avoir choisi l’option « contacter le propriétaire », pour un logement donné, le système vérifie si le l’utilisateur est connecté. Si ce n’est pas le cas le système amène l’utilisateur à s’authentifier avant de continuer. Une fois l’utilisateur connecté, le système invite l’utilisateur à insérer les informations de son compte mobile money ou flooz pour le retrait d’une somme convenue. Après la saisie des informations le système vérifie la validité des informations et lorsque celles-ci sont les bonnes la transaction est faite. Une nouvelle instance de prise contact est alors enregistrer en base de données et le système affiche au client toutes les informations nécessaires au client pour une prise de contact physique avec le propriétaire.

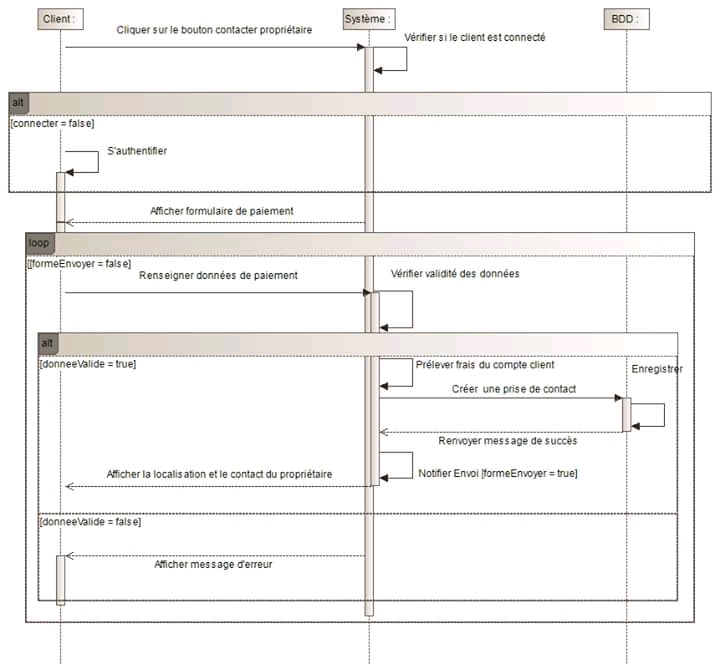


Figure 5: Diagramme de séquence du cas "contacter un propriétaire"

### 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 fini. 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 correspondent aux différentes situations se produisant lors de la réalisation de chaque cas d’utilisation.

Voici illustrer ci-dessous quelques diagrammes d’état transition de Hohaya

* Diagramme d’état transition du cas « Publier un logement »

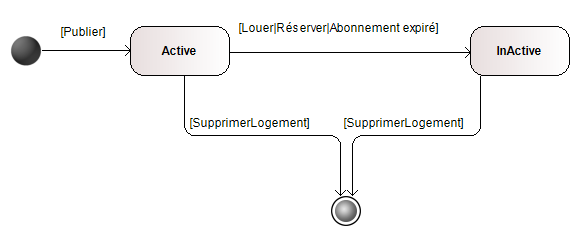


Figure 6 : Diagramme d'état transition de l’objet publication

* Diagramme d’état transition du cas « **Réserver un logement** »

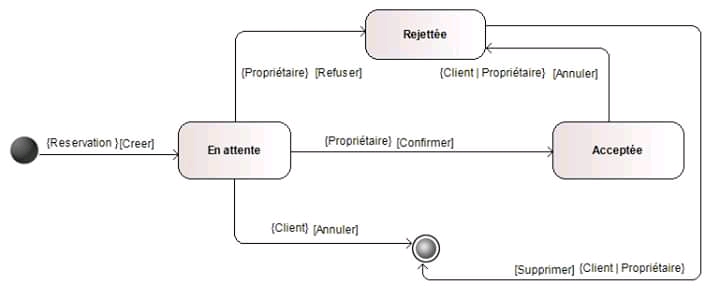


Figure 7: Diagramme d'état transition de l’objet réservation

### Diagramme d’activité

Le diagramme d'activité est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (multi-threads ou multi-processus). Le diagramme d'activité est également utilisé pour décrire un flux de travail (workflow). Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné [2].

# **Implémentation du système**

# ****CONCLUSION****

# ****BIBLIOGRAPHIE & WEBOGRAPHIE****

[1] : <https://fr.wikipedia.org/wiki/Acteur_(UML)> – consulté le 17 février 2020.

[2] : <https://fr.wikipedia.org/wiki/Diagramme_d%27activité> – consulté le 20 février 2020.