**DOSSIER D’INSERTION**

***Préambule***

Le dossier d’insertion est la partie du rapport qui présente la structure d’accueil, son fonctionnement et les conditions dans lesquelles nous avons été accueillis ainsi que les sujets abordés pendant le stage.

***Aperçu***

**INTRODUCTION**

**I-ACCEUIL ET INTEGRATION**

**II-PRESENTATION DE LA STRUCTURE D’ACCEUIL**

**III-FONCTIONNEMENT DE L’ENTREPRISE**

**IV-RESSOURCES DE L’ENTREPRISE**

**CONCLUSION**

**INTRODUCTION**

Les Technologies de l’Information et de la Communication sont un secteur d’activité qui exige des compétences à la fois théoriques et pratiques. De ce fait, l’acquisition du savoir-faire et l’acclimatation aux exigences de la vie professionnelle sont nécessaires pour tout ingénieur. C’est en réponse à ces attentes que nous avons présenté notre candidature auprès de l’entreprise **INTERNIS SARL** pour un stage académique, et avons eu le privilège d’être retenu. Dans cette partie, nous présenterons les étapes de notre insertion en son sein.

**I-ACCEUIL ET INTEGRATION**

Reçu comme stagiaire à INTERNIS SARL, nous avons débuté le stage le lundi 05 juillet 2021 des 08h30min, où un accueil chaleureux nous a été réservé par notre encadrant professionnel Monsieur Victoire KANKO. Nous avons tenu un bref entretien dans le but de savoir nos motivations, notre cursus, notre savoir-faire, de même ce que nous pouvons apporter à son organisation tout en mentionnant que durant le déroulement de notre stage, nous sommes appelés à être polyvalents aux vues des différentes activités de ladite structure. Ce dernier nous a aussi donné les attitudes à avoir à savoir : la ponctualité ; l’assiduité ; la disponibilité et l’envie d’apprendre.

**II-PRESENTATION DE LA STRUCTURE**

**D’ACCEUIL**

**II-1 Historique**

Créée en 2017 avec pour objet d’offrir aux entreprises et en particulier au camerounais des solutions innovantes et efficaces enfin que ces derniers puissent développer leurs activités et améliorer leur présence sur internet, INTERNIS est une entreprise de prestation de services informatiques et productions d’applications spécialisées. Elle fournit des solutions informatiques innovantes et propose à ses clients, un service de proximité qui rime avec compétence, expertise et sérieux en fonction de leurs besoins réels.

**II-2 Situation géographique**

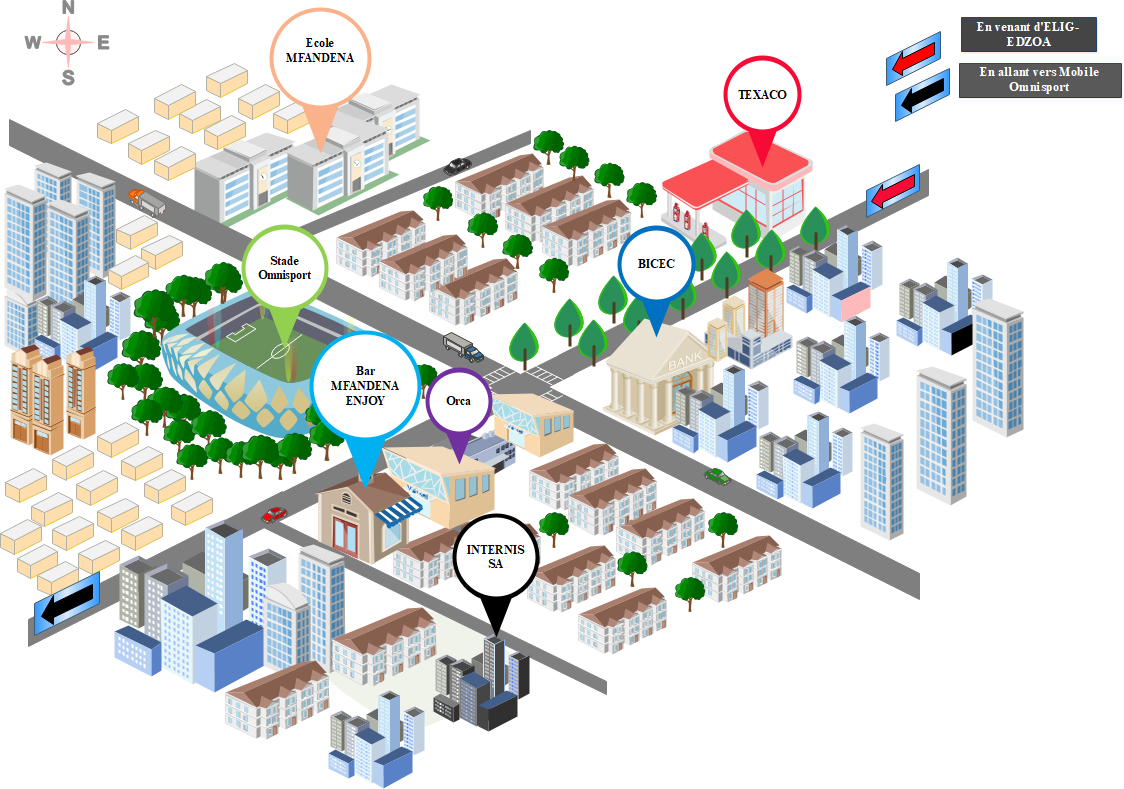
Les locaux de INTERNIS se trouvent à Yaoundé, au lieudit Mobile Omnisport

Figure 1 Plan de localisation de INTERNIS SARL

**II-3 Missions**

INTERNIS a pour mission principale l’amélioration du quotidien des citoyens camerounais à travers des solutions informatiques personnalisées. A cet effet, ses missions sont :

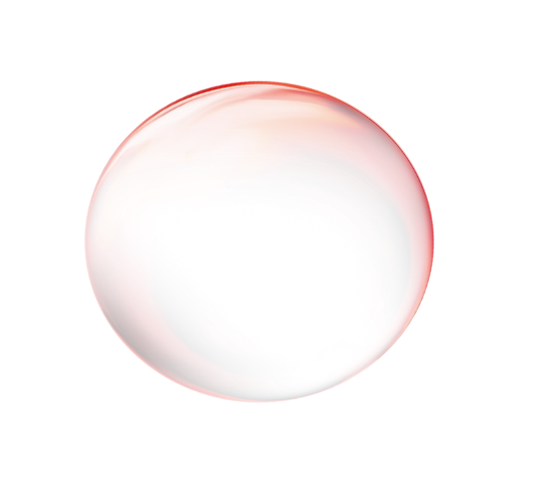
* **Sur le plan culturel** : promouvoir la culture africaine en général et camerounaise en particulier à travers les réseaux sociaux ;
* **Sur le plan social** : améliorer le quotidien des peuples en leur apportant des solutions personnalisées pour leur faciliter la vie ;
* **Sur le plan économique** : devenir un pôle d’excellence et participer tant que faire se peut à l’accroissement de l’économie camerounaise ;
* **Sur le plan technologique** : l’innovation technologique est un principe phare des objectifs à court et à long terme de INTERNIS c’est pourquoi elle se positionne tel un pilier de création et d’adaptation des solutions informatiques.

**II-4 Activités**

La création de l’entreprise résulte de la volonté de ses fondateurs, de mettre à la disponibilité des organisations privées et/ou publiques, nationales et internationales, un pôle d’expertise prouvée dans le secteur des TIC. Cette dernière offre des prestations de qualité, basées sur des idées innovantes, dans le but de les accompagner et de les conseiller dans le développement de leurs systèmes d’information. Dès lors, avec une expérience de plus en plus avancée dans le secteur des systèmes d’information, l’entreprise est structurée en plusieurs pôles à savoir :

* **Le pôle consulting** : assiste les clients dans la maitrise d’ouvrage des projets IT stratégiques ;
* **Le pôle Software Engineering :** éditeur du portail de gestion intégrée en temps réel ;
* **Le pôle Système et Infrastructures :** Architecte et infogérant d’infrastructures d’édition, de stockage, serveurs, réseaux et télécoms basées sur les dernières technologies.

**II-5 Quelques partenaires**



L’entreprise depuis sa création entretient des relations de partenariat avec les institutions telles que :

**GOOD FOOD**

**KAPT MEDIA**

**ELYO MEDIA**

**INTERNIS SA**

**ACRYMO**

Figure 2 Partenaires de INTERNIS SARL

**III-FONCTIONNEMENT DE L’ENTREPRISE**

La structuration de INTERNIS est relativement simple. Elle est structurée tel qu’il suit :

* **La Direction Générale** : coordonne les activités menées au sein de l’entreprise ;
* **Le Service Administratif** **et financier** : assure à la demande hiérarchique, la totalité du traitement d’informations particulières ;
* **Le Service Technique** : c’est le service qui gère tout ce qui est configuration, installation et logiciel, lors de la vente d’un matériel ou logiciel à un client et assure la mise en œuvre de nouvelles solutions informatiques ;

Direction Générale

Direction Technique

Département Génie Logiciel

Département Système et Reseaux

Direction Administrative et Financière

Comptabilité

Figure 3 Organigramme de INTERNIS SARL

**IV-RESSOURCES DE L’ENTREPRISE**

**IV-1 Ressources humaines**

* Personnel de maitrise
* 02 Ingénieurs de conception
* 02 Analystes programmeurs
* 03 Infographes
* Personnel d’appui
* 01 Comptable
* 01 coursiers

**IV-2 Ressources matérielles**

Dans l’exercice de ses activités, INTERNIS SARL à sa disposition un certain nombre d’équipements qui sont représentés dans le tableau suivant :

Tableau 1 Ressources matérielles de INTERNIS SARL

|  |  |  |
| --- | --- | --- |
| **Equipement** | **Libellé** | **Quantité** |
|  | Ordinateurs de bureau HP et DELL | **12** |
|  | Ordinateurs portables identiques (marque HP,1 Téra de DD, écran LED de 15 pouces, 4G de ram) | **2** |
|  | Imprimantes HP color LaserJet CP1215 | **2** |
|  | Photocopieuse | **1** |
|  | Relieuse | **1** |
|  | Routeurs CISCO | **1** |

**IV-3 Ressources logicielles**

Plusieurs logiciels sont utilisés dans le cadre professionnelle pour la réalisation des activités. Le tableau suivant représente les différents logiciels utilisés au sein de l’entreprise :

Tableau 2 Ressources logicielles de INTERNIS SARL

|  |  |
| --- | --- |
| Logiciel | Libellé |
|  | Un système d’exploitation d’un ordinateur de bureau : Windows  8.1 |
|  | Système d’exploitation des ordinateurs de bureau :  Windows 10 |
|  | Antivirus pour les ordinateurs de bureau :  Kaspersky + licence |
|  | Un système d’exploitation serveur : Debian 7 (Lenny) hébergeant le service Web |
|  | La suite office 2019 |

**CONCLUSION**

La phase d’insertion au sein d’INTERNIS SARL n’a présenté pour nous aucune difficulté majeure. Nous avons été chaleureusement accueillis, et accompagnés dans le plus grand professionnalisme. Nous nous sommes facilement imprégnés dans l’entreprise et au terme de cette phase, nous avons fait différentes observations qui nous donnaient déjà une imagination sur l’intitulé notre projet de stage.

**CAHIER DES CHARGES**

***Préambule***

Cette première étape de la phase technique de notre travail présente le contexte et les objectifs du projet, ainsi que les besoins, ressources et contraintes de celui-ci afin d’aboutir à un planning prévisionnel. La réalisation de ce document implique le maitre d’œuvre et d’ouvrage.

***Aperçu***

**INTRODUCTION**

**I-CONTEXTE ET JUSTIFICATION**

**II-OBJECTIFS DU PROJET**

**III-EXPRESSION DES BESOINS**

**IV- ESTIMATION DU COUT**

**V-PLANIFICATION**

**VI-CONTRAINTES**

**VII-LIVRABLES**

**CONCLUSION**

**INTRODUCTION**

Mettre en place une plateforme nécessite de toute évidence le respect d’un cahier des charges. Le cahier des charges est établi par le maitre d’œuvre en accord avec le maitre d’ouvrage. Ici, il est question de définir de manière précise les orientations du projet, de son contexte à ses livrables, en passant par ses objectifs, les besoins à satisfaire, les ressources à mobiliser, les contraintes à respecter et le planning prévisionnel.

**I-CONTEXTE ET JUSTIFICATION**

La gestion immobilière représente bien souvent un véritable casse-tête pour les propriétaires. Lorsqu’il s’agit de location, le manque de temps, les loyers impayés et la complexité des démarches fiscales et juridiques font généralement que ceux-ci se tournent vers des prestataires pour administrer leurs propriétés. Ce qui est non seulement peu sûr mais aussi très couteux.

Afin d’y trouver des solutions, il serait nécessaire de mettre à la disposition des propriétaires/bailleurs un outil grâce auquel ils pourront non seulement gérer de façon plus flexible leurs logements mais aussi où ils pourront être accompagnés par des professionnels et ceci à moindre coût.

C’est partant de ce constat qu’ici à INTERNIS nous avons pris l’initiative de mettre sur pied une plateforme de gestion immobilière, afin d’apporter une solution innovante pour satisfaire non seulement les propriétaires de biens immobiliers mais aussi les locataires

**II-OBJECTIFS DU PROJET**

Le projet que nous nous proposons de mener à terme vise la conception et la réalisation d’une plateforme de gestion immobilière. Cette plateforme permettra principalement d’accompagner les propriétaires de biens dans la gestion de leurs biens. Plus précisément, il d’agira de :

* Faciliter les démarches de gestion : Rédaction de baux, réalisation d’états des lieux, etc. ;
* Faciliter le prélèvement de loyers ;
* Automatiser l’édition et l’envoie des quittances de loyer ;
* Faciliter le traitement des loyers impayés ;
* Faciliter la gestion des factures.

**III-EXPRESSION DES BESOINS**

Notre solution devra répondre aux attentes des utilisateurs en matière de fonctionnalités. Ces attentes sont réparties selon deux axes et ont des déclinaisons selon que l’utilisateur est un locataire, un bailleur ou alors un administrateur.

**III.1 Besoins Fonctionnels**

Les besoins fonctionnels se rapportent aux fonctionnalités que l'application en question doit offrir pour satisfaire les utilisateurs. Les fonctionnalités que doit intégrer l'application à développer peuvent être décrites comme suit :

* + **La gestion compte utilisateur** :

Possibilité d’ajouter, de supprimer et de modifier un compte utilisateur ;

* + **La gestion des biens**

La possibilité pour le bailleur d’ajouter ses biens, de les modifier, de les supprimer, de fixer les loyers, etc. ;

* + **La gestion des locataires**

La possibilité pour le bailleur d’ajouter ses locataires ;

* + **La gestion des factures et des quittances** :

Le bailleur pourra consulter l’état de paiement des factures de chacun de ses locataires et réciproquement. Apres cela, et sous certaines conditions il pourra générer et envoyer au locataire sa quittance ;

* + **La gestion des inventaires** : Cette opération consistera à capturer l’état des équipements d’un logement à une date donnée (pour locations meublées) ;
  + **La gestion des états de lieux** : Cette opération consistera à capturer l’état d’un logement et de ses pièces à une date donnée (pour locations simples comme pour locations meublées ;
  + **La gestion des rénovations** : Ici, le locataire et bailleur pourront enregistrer les travaux faits (ou qui devront être faits) sur leur logement ;
  + **La gestion du contrat de bail** :

Par cette opération chaque bailleur pourra ajouter ou retirer des clauses dans ses contrats avec ses locataires ;

* + **La gestion des packages** :

Permettra de créer des privilèges et de ne les accorder qu’aux utilisateurs ayant payé une certaine somme.

**III.2 Besoins Non Fonctionnels**

Outre les besoins fonctionnels de notre application, d’autres besoins non indispensables ne sont pas à négliger. En effet, la satisfaction de ces besoins dits non fonctionnels améliore grandement l’application. Les besoins non fonctionnels de notre système sont multiples :

* Fiabilité : La plateforme doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante ;
* La convivialité : l’application doit être facile à utiliser ;
* L’ergonomie : La plateforme doit être adaptée à tout type d’utilisateur sans qu'aucun effort ne soit fourni par ce dernier ;
* La maintenance : Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation ;
* La performance : cette application doit être avant tout performante c’est-à-dire à travers ses fonctionnalités, doit répondre à toutes les exigences des utilisateurs d’une manière optimale ;
* Un système de newsletter : qui permettra d’informer les utilisateurs de la plateforme des différentes mises à jour effectuées sur celle-ci.

**IV-ESTIMATION DU COUT DU PROJET**

Tout projet, aussi petit qu’il soit, a un coût. Cependant, le coût d’une application n’est   
pas donné au hasard ; de nombreux facteurs sont pris en compte dans l’estimation de celui-ci.   
Pour déterminer le coût de notre application, nous allons recenser les ressources nécessaires à   
sa réalisation afin d’avoir une idée sur le coût d’un tel projet.

**IV-1 Ressources Matérielles**

Tableau 3 Ressources matérielles du projet

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Désignation | Caractéristiques | Fonction | Quantité | | Prix Unitaire | Prix Total (FCFA) |
| Ordinateur  portable | Processeur Intel Core i5,  CPU 2.5 GHz, HDD  500Go, Ram 8Go | Terminaux de travail | 01 | | 402 500 | 402 500 |
| Modem  WIFI | Modem internet wifi 4G | Terminal de  connexion internet | 01 | | 44 000 | 44 000 |
| Forfait internet | Bande passante | 03 | | 25 000 | 125 000 |
| Clé USB | 16 Go | Transfert et transport  de fichier | 01 | | 17 250 | 17 250 |
| Rame de  papier | Format A4 80G blanc | Prise de note et  brouillon | 01 | | 4 485 | 4 485 |
| Total | | | | **593 235** | | |

*Tous les prix fournis dans ce tableau sont tous issus de la Mercuriale des prix 2020*

**IV-2 Ressources Logicielles**

Tableau 4 Ressources Logicielles du projet

|  |  |  |
| --- | --- | --- |
| Désignation | Fonction | Prix (FCFA) |
| MICROSOFT WINDOWS 10 | Système d’exploitation | 126 000 |
| MICROSOFT OFFICE 2016  PROFESSIONNEL | Suite bureautique | 177 000 |
| ASTAH COMMUNITY | Outil de modélisation | 659 920 |
| ADOBE PHOTOSHOP CC 2017 | Outil de graphisme | 532 245 |
| GOOGLE  CHROME | Navigateur | Gratuit |
| GANTT PROJECT | Outil de planification de projet | Gratuit |
| MYSQL | SGBD | Gratuit |
| VISUAL STUDIO CODE | Outil de développement | Gratuit |
| TOTAL | | **1 495 165** |

*Les prix des logiciels Windows 10 et Office 2016 sont issus de la Mercuriale des prix 2020. Le prix d’Astah Community   
provient de https://www.fadace.developpez.com/astah ; Photoshop prend son prix sur   
https://www.linformaticien.com/actualites/id/24591/adobe-tous-les-prix-de-la-suitecc2017.aspx*

**IV-3 Ressources Humaines**

Tableau 5 Ressources Humaines du projet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Poste | Nombre de participants | Prix par jour | Nombre de jour | Total (FCFA) |
| Maitre d’œuvre  Chef de projet | 01 | / | / | 2 000 000 |
| Analyste concepteur | 01 | 100000 | 24 | 2 400 000 |
| Programmeur | 01 | 150000 | 21 | 3 150 000 |
| Testeur | 02 | 30000 | 5 | 150 000 |
| TOTAL | | | | **7 700 000** |

*Source : Benchmaking (Etude du marché)*

**IV-4 Cout total du projet**

Tableau 6 Cout total du projet

|  |  |
| --- | --- |
| Désignation | Montant (FCFA) |
| Ressources matérielles | **593 235** |
| Ressources logicielles | **1 495 165** |
| Ressources humaines | **7 700 000** |
| Charges (5% des ressources) | **489 420** |
| Imprévus (10% ressources) | **978 840** |
| TOTAL | **11 256 660** |

**V-PLANIFICATION DU PROJET**

Il est judicieux avant d’entamer notre projet d’établir, en plus d’une liste de compétences et ressources requises, un emploi du temps des différentes étapes qui mèneront à son aboutissement. Ces étapes sont :

Tableau 7 Planification du projet

|  |  |  |
| --- | --- | --- |
|  | Tâches | Durée (jours) |
| INTEGRATION | **Insertion en entreprise** | **7** |
| **Elaboration du cahier des charges** | **14** |
| DEVELOPPEMENT | **Analyse et conception** | **24** |
| **Codage** | **21** |
| DEPLOIEMENT | **Tests unitaires,**  **d’intégration** | **5** |
| **Correction et validation** | **2** |
| **Déploiement** | **2** |
| EXPLOITATION | **/** | **Indéterminée** |

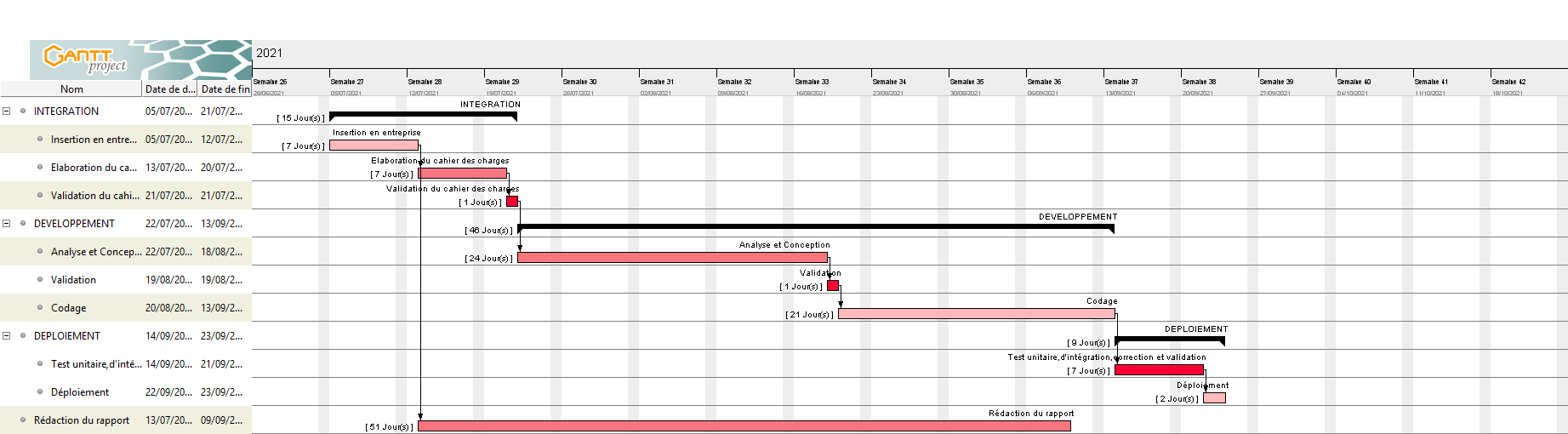
La rédaction du rapport s’effectuera parallèlement à ces tâches afin d’assurer la cohérence et d’économiser en temps.

Figure 4 Diagramme de GANTT

D’après cette première estimation nous devons donc réaliser le projet dans un délai de 68 jours ouvrables à compter de la date de validation du thème.

**VI-CONTRAINTES DU PROJET**

Pour mettre en place notre plateforme, nous devrons respecter 3 contraintes à savoir :

* Les contraintes de coût : Le budget fixé pour la réalisation de l’application devra   
  être respecté conformément aux prévisions c’est-à-dire sans surenchère ou sous-  
  enchère ;
* Les contraintes de délai : La réalisation de ce système devra respecter les   
  contraintes de temps fixés et donc être faite sur une durée de 03 mois et   
  devra respecter les objectifs fixés ;
* Les contraintes de qualités : Après avoir fixé le prix et un délai, nous devons   
  produire une application flexible, réutilisable et surtout évolutive.

**VII-LIVRABLES**

Dans cette section, il sera question pour nous de présenter les différents éléments qui vous serons donnés à la fin du projet. Il s’agit notamment :

* D’un rapport complet comportant :
* Le dossier d’insertion
* Le cahier des charges
* Le dossier d’analyse
* Le dossier de conception
* Le dossier de réalisation
* Le guide d’utilisation
* D’un CD-ROM Comprenant :
* Le code source de l’application

**CONCLUSION**

Il était question pour nous dans cette partie de réaliser le cahier des charges. Ce dernier nous a permis de mieux cerner les besoins des utilisateurs et de prendre connaissance des résultats attendus de notre travail. Il est donc question de passer à la phase d’analyse du système.

**DOSSIER D’ANALYSE**

***Préambule***

Le dossier d’analyse a pour but d’affiner le choix de la méthode à utiliser tout au long du projet. Il permet dans la même lancée de faire comme son nom l’indique une analyse détaillée du projet à mettre en œuvre.

***Aperçu***

**INTRODUCTION**

**I-PRESENTATION DE L’EXISTANT**

**II-CRITIQUE ET LIMITES**

**III-PROBLEMATIQUE**

**IV-PROPOSITION DE SOLUTION**

**V-PRESENTATION DE LA DEMARCHE D’ANALYSE**

**VI-MODELISATION DE LA SOLUTION**

**CONCLUSION**

**INTRODUCTION**

L’analyse est la décomposition d’un problème en ses différents éléments en vue d’un traitement informatique. De ce fait pour la bonne réalisation d’un projet, il est très important d’effectuer une bonne analyse afin d’atteindre les objectifs spécifiés dans le cahier des charges. La réalisation d’un bon projet repose principalement sur son analyse. Ainsi, un projet mal analysé ne saurait donner de bons résultats.

**I-PRESENTATION DE L’EXISTANT**

Quand on réalise un investissement immobilier locatif, la recherche et l’achat du bien à louer ne représentent que la première étape. La seconde consiste à louer le logement pour générer des revenus. Cette seconde étape, déterminante pour la rentabilité du projet, comporte deux missions : la mise en location et la gestion locative. Par gestion locative, on entend ici l’ensemble d’étapes qui interviennent juste après la mise en location, c’est-à-dire une fois que le candidat soit devenu locataire, et pour qu’un candidat soit considéré comme locataire, il faut que :

* Le bail (ou contrat de location) soit signé
* Le [dépôt de garantie](https://proprietaire.monbelappart.com/fonctionnement-depot-garantie/) soit versé
* [L’état des lieux d’entrée](https://proprietaire.monbelappart.com/etat-des-lieux-entree/) soit effectué et signé par les deux parties
* Les clés du logement soient confiées au locataire

C’est seulement à ce moment que le locataire est considéré comme tel. Une fois ces prérogatives remplies, on entre dans la phase de gestion locative. Par simplicité, beaucoup de propriétaires bailleurs préfèrent se tourner vers une agence pour gérer la [location](https://bailpdf.com/guides/location) de leur bien. Or, Le monde de l’immobilier se tourne de plus en plus vers le numérique. Après les banques et les assurances en ligne, on trouve aujourd’hui des agences de gestion locative 100 % en ligne. Ces professionnels de l’immobilier offrent les mêmes services que les agences immobilières classiques tout en tirant au maximum profit de la flexibilité offerte par les solutions digitales. Actuellement, les principales plateformes de gestion locatives sont :

Tableau 8 Plateformes de gestion locative existantes

|  |  |  |
| --- | --- | --- |
| Services proposés | Nom | logo |
| Administration simplifiée, modèle de document, tenue de compte, révision de loyer, déclaration fiscale etc. | GERANT-IMMO  (www.gerant-immo.com) |  |
| Gestion des comptes, Système de messagerie, Quittances électroniques, Révision de loyer, Forum de discussion | RENTILA  (www.rentila.com) |  |
| Vente de logement ou de terrain  Location non meublée  Location meublée | IMMO Cameroun  (www.immo-cameroun.cm) |  |

**II-CRITIQUE DE L’EXISTANT**

Après avoir effectué une étude minutieuse du fonctionnement des plateformes ci-dessus cités, nous constatons que chacune d’elle possède des avantages et des insuffisances notoires. Le tableau qui suit mettra en exergue les difficultés qu’impose le système classique de gestion des biens immobiliers ainsi que les éventuelles solutions proposées dans le cadre de notre projet.

Tableau 9 Critique de l'existant

|  |  |  |
| --- | --- | --- |
| INSUFFISANCES | CONSEQUENCES | SOLUTIONS |
| Ces plateformes ne sont pas très adaptées aux besoins des camerounais, en termes d’administration, de documents légaux et de charges | Impossibilité pour ceux-ci d’administrer complètement leur bien de façon automatique | Création d’une plateforme de gestion locative qui répondra principalement aux besoins des camerounais |
| Les utilisateurs ne sont pas informés avant l’expiration de leurs baux, ou de la date de paiement de leurs loyers | Les propriétaires et locataires se retrouvent alors dans des conflits liés au manque d’informations sur les dates de paiement des frais de locations | Création d’un système de notification automatique en ce qui concerne toutes les transactions liées à un bien avec des fréquences de notification prédéfinies |
| En cas d’incident ou lorsqu’un utilisateur veut effectuer une rénovation sur un bien il n’est pas accompagné | L’utilisateur devra se déplacer, à la recherche d’un technicien qualifié pour résoudre son problème | Permettre aux utilisateurs de trouver des techniciens qualifiés sans forcément se déplacer |

**III-PROBLEMATIQUE**

L’étude que nous avons menée plus haut nous a permis de constater un certain nombre de manquements sur les systèmes en place. Notre objectif étant d’apporter une solution à ces différentes insuffisances nous nous sommes posés la question de savoir : comment faciliter complètement le processus de gestion locative, allant de la rédaction du bail à la résolution des incidents ?

**IV-PROPOSITION DE SOLUTION**

Après avoir observé plusieurs manquements dans le fonctionnement des plateformes de gestion existantes, il apparait clairement qu’au Cameroun, une autre approche doit être adoptée pour la mise en place d’une nouvelle forme de gestion locative. Une application ouverte à tous, devra combler toutes les faiblesses rencontrées avec les systèmes classiques. C’est dans cette optique que nous envisageons concevoir une plateforme de gestion immobilière dans laquelle la gestion des biens immobiliers et des locations sera plus adaptée pour les utilisateurs locaux et qui permettra entre autres de suivre les loyers, de générer les documents légaux relatifs aux locations (selon la législation camerounaise), de générer les quittances, de gérer les travaux et interventions et de mettre en relation le bailleur et les locataires.

**V-PRESENTATION DE LA DEMARCHE D’ANALYSE**

L’analyse est une étape fondamentale dans la conception d’un logiciel. C’est la base de tout travail de réalisation de systèmes d’information. Un système d’information est un système organisé de ressources, de personnes et de structures qui évoluent dans une organisation et dont le comportement coordonné vise à atteindre un but commun.

Plusieurs méthodes et langages ont étés développés pour faciliter et normaliser l’analyse et la conception des systèmes d’information parmi lesquels nous avons principalement UML et MERISE.

MERISE (Méthode d’Etude et de Réalisation Informatique pour les Systèmes d’Entreprise) est une méthode d’analyse, de conception structurelle et de réalisation des systèmes d’informations très utilisées notamment dans les entreprises françaises. Elle est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. Son but principal est d’arriver à concevoir un système d’information (SI).

UML (Unified Modeling Language) quant à lui, est un langage de modélisation des systèmes standard, qui utilise des diagrammes pour présenter chaque aspect d’un système en s’appuyant sur la notion d’orienté objet qui est un véritable atout pour ce langage. UML propose donc une approche différente de celle de MERISE en ce sens qu’il associe les données aux traitements. En effet, avec UML centraliser les données d’un type et les traitements associés permet de limiter les points de maintenance dans le code et de faciliter l’accès à l’information en cas d’évolution du logiciel. De plus, UML décrit la dynamique du système d’information comme un ensemble d’opérations attachées aux objets du système.

**V-1 Comparaison entre UML et MERISE**

Tableau 10 Comparaison entre UML et MERISE

|  |  |
| --- | --- |
| MERISE | UML |
| MERISE utilise une approche systémique. | UML est un langage de modélisation objet à qui il faut associer un processus (2TUP, RUP, XUP) pour en faire une méthode |
| MERISE propose de considérer le système réel selon deux points de vue : • Une vue statique (données) • Une vue dynamique (traitements) | UML propose une approche différente en ce sens qu’il associe les données et les traitements. |
| Avec la méthode MERISE, nous avons une étude séparée des données et des traitements | Avec UML, centraliser les données d’un type et les traitements associés permet de limiter les points de maintenance dans le code et facilite l’accès à l’information en cas d’évolution |
| Démarche « Bottom Up » : De la base de données vers le code | Démarche « Top Down » : Du modèle vers la base de données |

**V-2 Justification de la démarche choisie**

Au regard de la complexité croissante des systèmes d’informations et de cette analyse comparative, UML s’avère être la meilleure démarche d’analyse pour ce projet. UML est un langage unifié de modélisation qui permet de spécifier, visualiser, construire et documenter de manière claire et précise un système d’information. Le modèle fourni par UML est valable pour n'importe quel langage de programmation et grâce à ses diagrammes, il offre une description statique, dynamique et fonctionnelle du système à réaliser. Nous avons précédemment fourni une description brève d’UML, la section qui suivra présentera de façon un peu plus détaillée le langage UML. Dans le cadre de ce projet, nous travaillerons avec les spécifications de la version 1.4 sortie en 2003.qui compte 9 diagrammes repartis sur 2 vues.

* Vue statique :
* Diagrammes de cas d'utilisation ;
* Diagrammes d'objets ;
* Diagrammes de classes ;
* Diagrammes de composants ;
* Diagrammes de déploiement ;
* Vue dynamique :
* Diagrammes de collaboration ;
* Diagrammes de séquence ;
* Diagrammes d'états-transitions ;
* Diagrammes d'activités ;

Malgré tous ses atouts incontestés, UML ne propose néanmoins pas de façon claire une démarche à suivre dans l’élaboration d’un projet, de la compréhension des besoins des utilisateurs à la production du logiciel. Pourtant, le choix d’une méthode est d’une importance capitale dans la démarche de mise sur pied d’un logiciel, c’est pourquoi, il faut lui associer un processus de traitement. Parmi les méthodes de conception s’appuyant sur UML (2TUP, RUP, XUP), notre choix s’est porté sur 2TUP dont le point suivant fera l’objet de sa présentation.

**V-3 Présentation du processus choisi**

La conception d’un système d’information passe par une séquence d’étapes ordonnées dont l’objectif est de produire un logiciel répondant aux besoins des utilisateurs dans le temps et des coûts prévisibles. Or UML est un langage qui ne propose que des diagrammes. Pour concevoir un SI avec lui, il est impératif de lui associer un processus.

2TUP (Two Tracks Unified Process) permet la production d’un logiciel de haute qualité avec un temps et un budget limité. C’est une méthode de développement de logiciel qui s’appuie sur UML et définit QUI fait QUOI, QUAND et COMMENT pour atteindre un objectif.

2TUP est un processus de développement logiciel :

* **Générique** : Il est nécessaire d'adapter le processus unifié au contexte d'un projet, d'une équipe, d'un domaine ou d'une organisation. Il existe donc un certain nombre de méthodes issues d’UP : 2TUP, RUP, XUP, EUP, AUP…
* **Itératif et incrémental** : Le projet est découpé en itérations de courtes durées (environ 1 mois) qui permettent de mieux suivre l’avancement global. A la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.
* **Centré sur l’architecture logicielle** : En effet tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.
* **Piloté par les risques** : Les risques majeurs du projet doivent être identifiés au plus tôt mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l’ordre des itérations.
* **Conduit par les cas d’utilisation** : Le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d’utilisation du futur système sont identifiés, décrits avec précision et priorisés.

La méthode 2TUP propose un cycle de développement en « Y », qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle des deux branches à savoir la branche fonctionnelle (étude de l’application) et la branche technique (étude de l’implémentation) pour aboutir à la phase de réalisation.

2TUP étant un processus associé au langage UML, il utilise pour chacune de ces branches des diagrammes d’UML comme suit :

* **Branche technique** : Cette branche capitalise le savoir-faire et les contraintes techniques. Elle réduit les risques de produire un système inadapté aux utilisateurs. Elle possède deux diagrammes : le diagramme de classe et le diagramme de composants
* **Branche fonctionnelle** : Elle capitalise la connaissance du métier de l’entreprise et produit un modèle des besoins focalisé sur le métier des utilisateurs. Trois diagrammes sont utilisés dans cette branche à savoir : le diagramme des cas d’utilisation, le diagramme de séquence et le diagramme d’activité
* **Branche de réalisation** : C’est le point de réunion des deux branches. Elle permet de mener une conception applicative afin de livrer d’une solution adaptée aux besoins spécifiés par les utilisateurs. Le diagramme que nous utilisons dans cette branche est le diagramme de déploiement.

La figure suivante est une illustration du processus 2TUP :

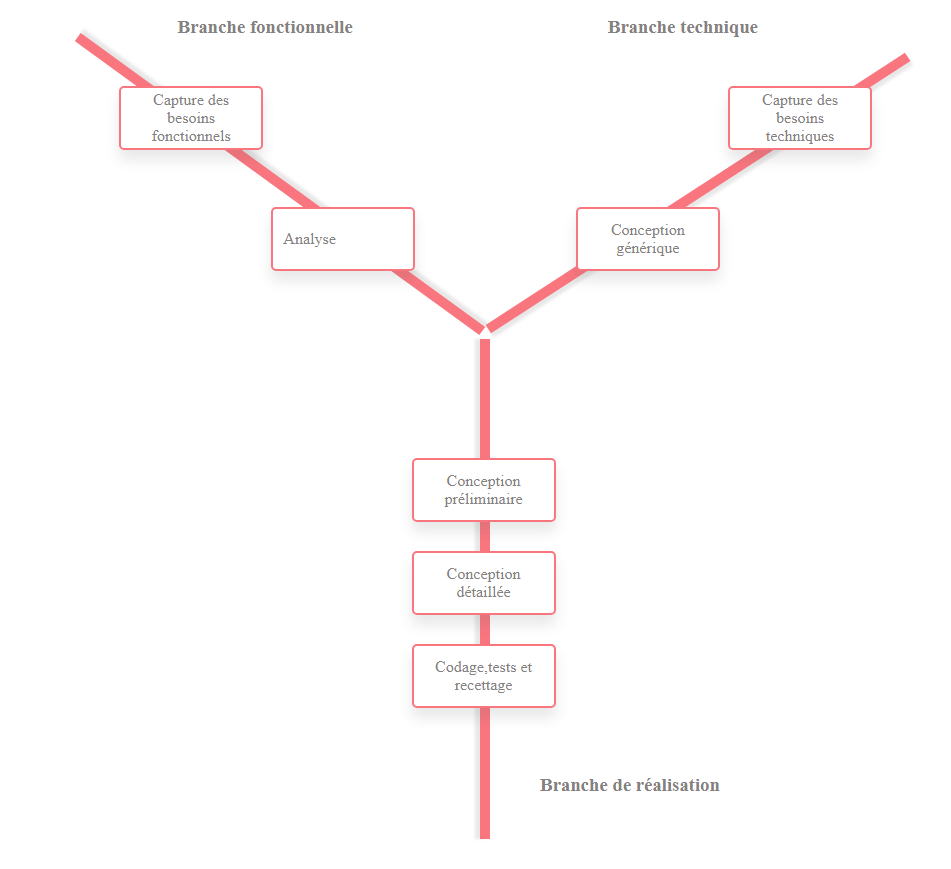


Figure 5 Présentation du processus 2TUP

**VI-MODELISATION DE LA SOLUTION**

**VI-1 Diagramme de cas d’utilisation**

1. **Présentation**

Les diagrammes de cas d’utilisation identifient les fonctionnalités fournies par le système (cas d’utilisation), les utilisateurs qui interagissent avec le système (acteurs), et les interactions entre ces derniers. Les cas d’utilisation sont utilisés dans la phase d’analyse pour définir les besoins de haut niveau du système. Les cas d’utilisation se prolongent au-delà des diagrammes imagés. En fait, des descriptions textuelles des cas d’utilisation sont souvent employées pour compléter ces derniers et représentent leurs fonctionnalités plus en détail.

1. **Formalisme**

Les principaux éléments constitutifs d’un diagramme de cas d’utilisation sont : les acteurs, les cas d’utilisation et les relations entre eux.

Tableau 11 Formalisme du diagramme de cas d'utilisation

|  |  |  |
| --- | --- | --- |
| **Elément du diagramme** | **Description** | **Représentation graphique** |
| **L’acteur** | Toute entité qui interagit avec le système |  |
| **Le cas d’utilisation** | Fonctionnalité réalisable par le système |  |
| **L’association** | Lien entre un acteur et les cas d’utilisation auxquels il a accès |  |

Il est possible de mettre en relation des acteurs pour traduire l’héritage, ou de mettre en relation des cas d’utilisation pour traduire extension, inclusion, ou généralisation.

Tableau 12 Types de relation dans un diagramme de cas d'utilisation

|  |  |  |
| --- | --- | --- |
| Type de relation | Description | Représentation graphique |
| Inclusion | Un cas A est en inclusion dans un cas B si l’exécution du cas A passe obligatoirement par l’exécution du cas B |  |
| Extension | Un cas A étend un cas B si après avoir réalisé le cas B, on peut réaliser le cas A |  |
| Généralisation des cas d’utilisation | Un cas A est une généralisation d’un cas B si le cas B est une autre manière d’effectuer le cas A (spécialisation) |  |
| Généralisation des acteurs | Un acteur A hérite d’un acteur B si en plus de réaliser ses cas d’utilisation, réalise aussi ceux de l’acteur B |  |

1. **Identification des acteurs**

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l’état du système, en émettant et/ou en recevant des messages susceptibles d’être porteurs de données.

Dans le cas de notre application et suivant le module « Gestion des comptes utilisateurs », nous avons pu identifier plusieurs acteurs parmi lesquels :

* **Le visiteur :** Il représente tout utilisateur qui n’a pas encore de compte dans notre plateforme. A cet effet, il peut juste créer un compte ;
* **Le bailleur** : Il a la possibilité de gérer ses biens, ses locations et ses locataires. Il peut consulter l’état de paiement des factures de ses locataires, ajouter des clauses dans son contrat de bail, etc. ;
* **Le locataire :** Il a la possibilité de consulter ses locations, ses factures et les quittances y afférentes. Il peut payer ses factures, consulter ses contrats, signer ses contrats, etc. ;
* **L’administrateur** : Il est le seul à avoir l’information complète du système. Il peut faire la gestion globale du système comme par exemple sur un utilisateur, modifier et supprimer un compte.

1. **Identification des cas d’utilisation**

Cette étape nous permet de ressortir une manière d’utiliser le système et permet d’en décrire les exigences fonctionnelles. Toujours suivant notre module à réaliser voici les cas d’utilisation retenues :

Tableau 13 Identification des acteurs et des cas d'utilisation

|  |  |
| --- | --- |
| Acteurs | Cas d’utilisation |
| Visiteur | S’inscrire |
| Bailleur | * Gérer mes biens * Gérer mes locataires * Gérer mes états de lieux * Gérer mes inventaires * Consulter mes articles * Consulter mon compte * Gérer mes rénovations * Consulter mes factures * Consulter mes quittances * Gérer mes locations * Signer des documents * Noter des artisans * Souscrire à des packages |
| Locataire | * Consulter son compte * Gérer mes rénovations * Consulter mes factures * Consulter mes quittances * Gérer mes locations * Signer des documents * Noter des artisans * Souscrire à des packages |
| Administrateur | * Gérer les comptes utilisateur * Gérer les package * Gérer les artisans |

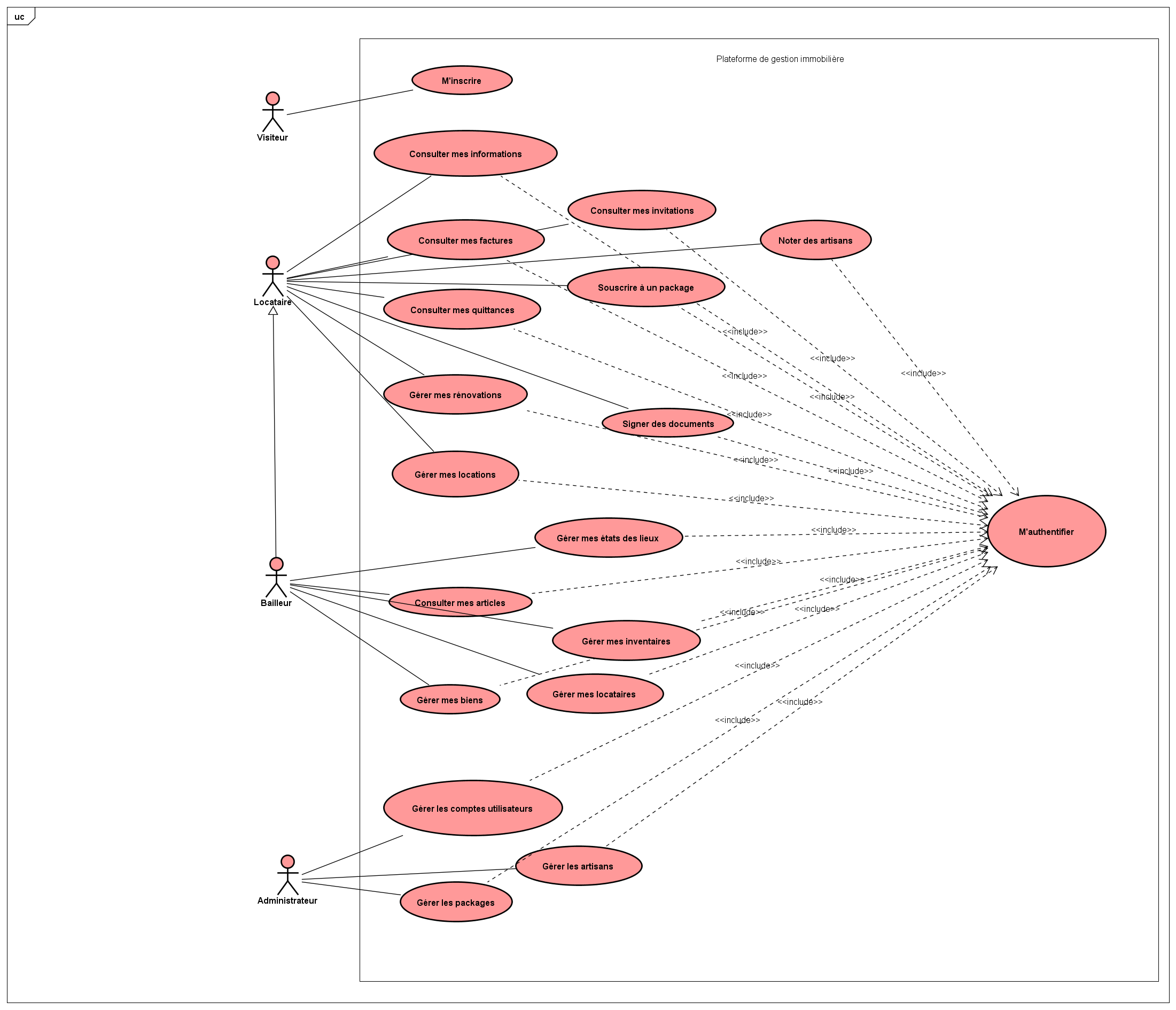
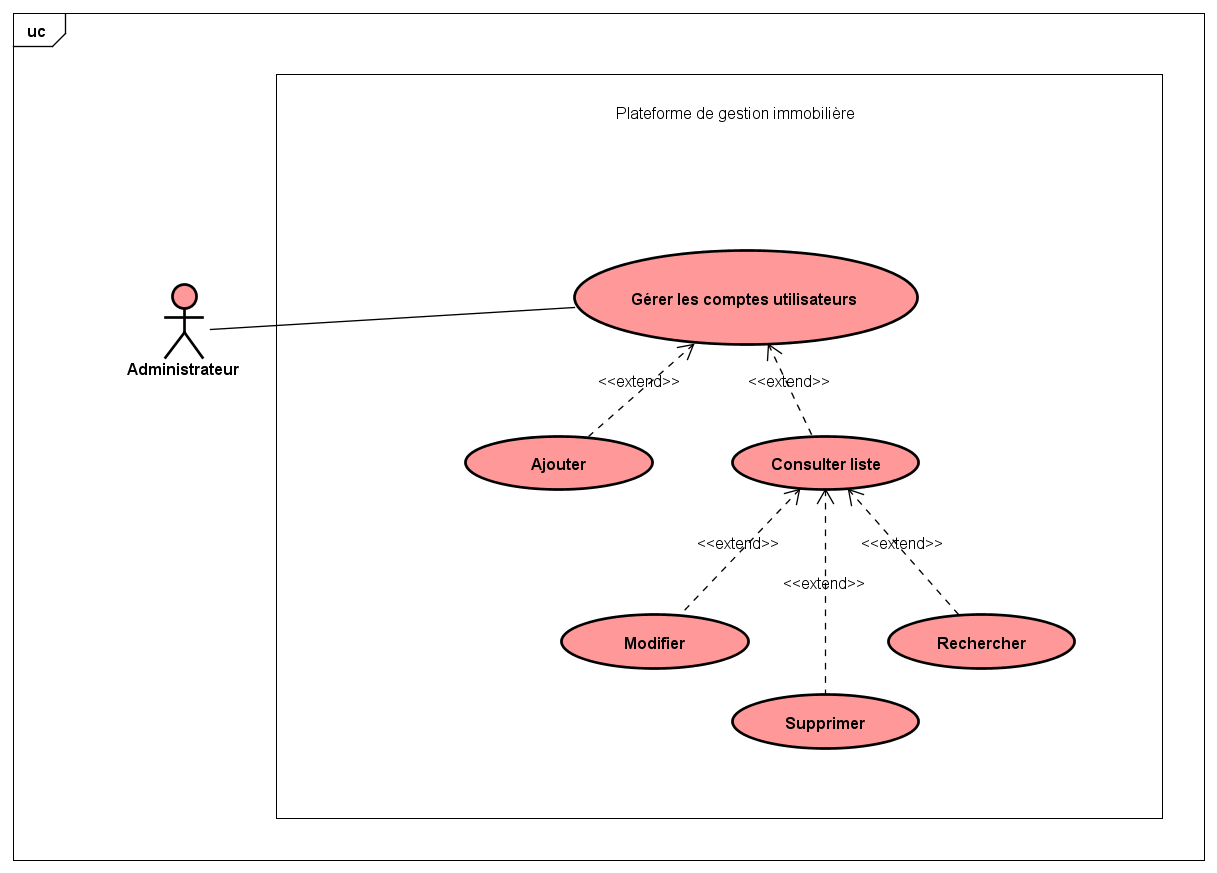
1. **Représentation graphique de quelques cas d’utilisation du système**
2. **Diagramme de cas d’utilisation global du système**

Figure 6 Diagramme de cas d'utilisation global du système

1. **Diagramme de cas d’utilisation relatif à la gestion des comptes utilisateur**

Figure 7 Diagramme de cas d’utilisation relatif à la gestion des comptes utilisateur



1. **Diagramme de cas d’utilisation relatif à la gestion des biens**

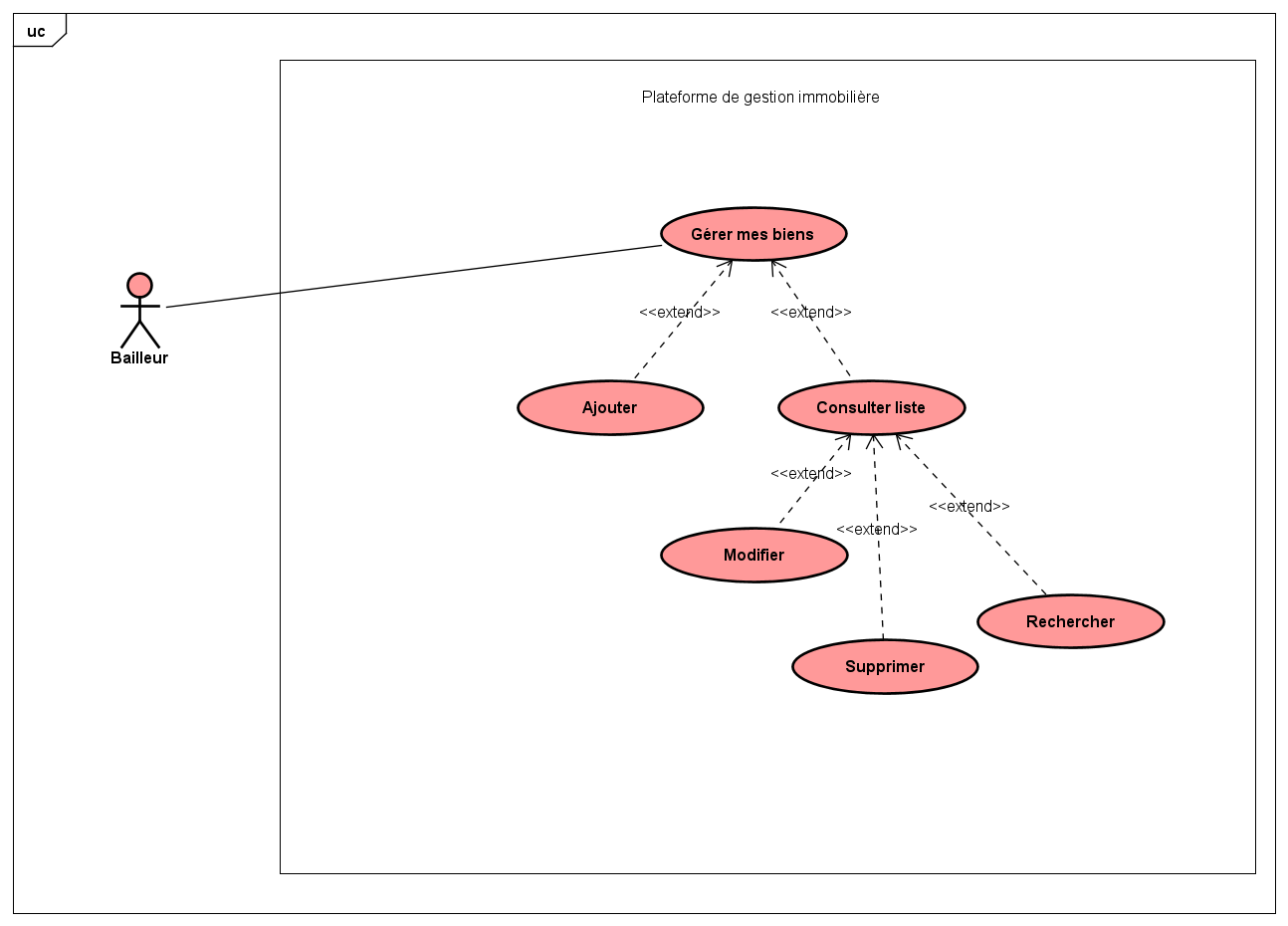


Figure 8 Diagramme de cas d’utilisation relatif à la gestion des biens

1. **Diagramme de cas d’utilisation relatif à la gestion des locataires**

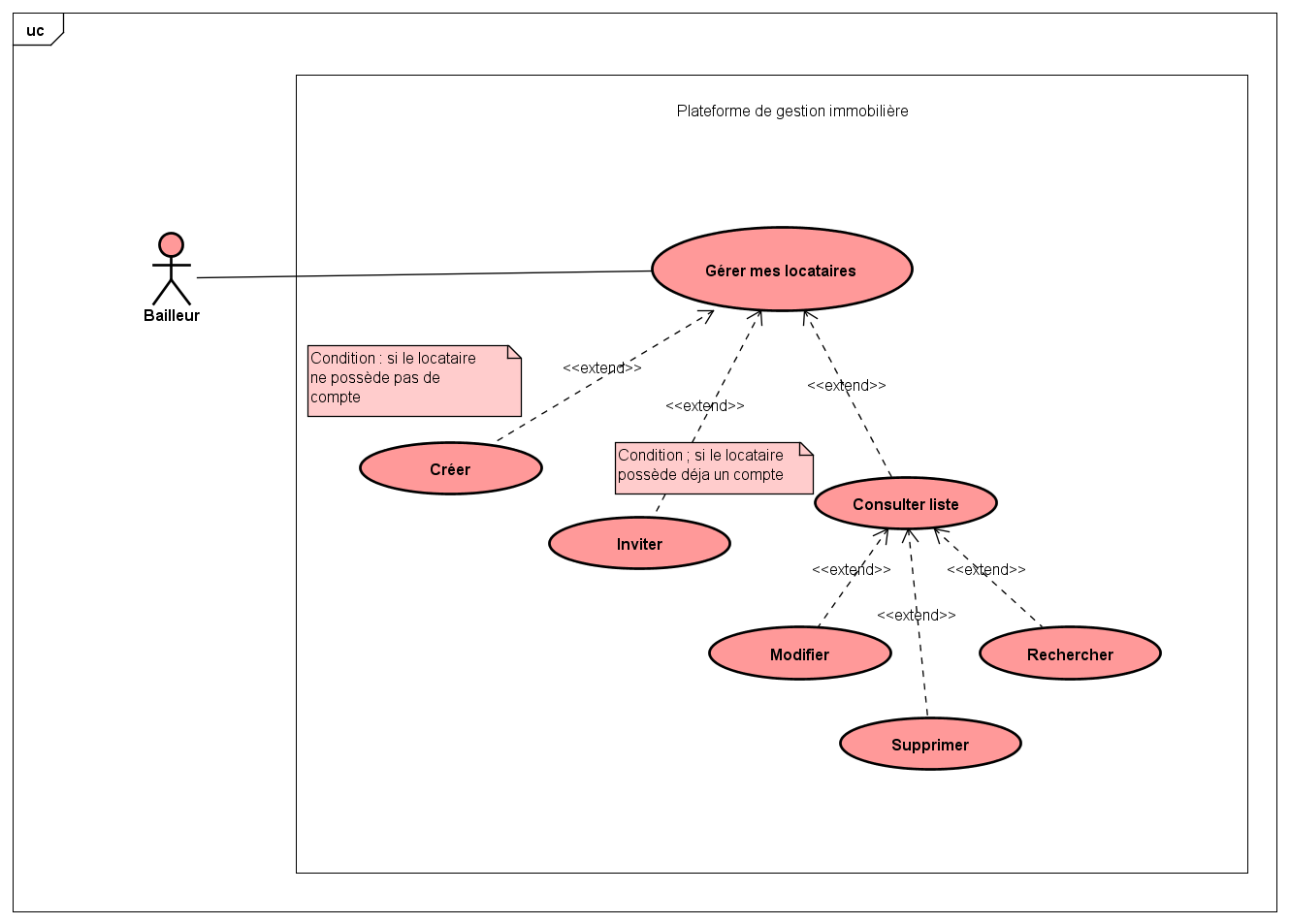


Figure 9 Diagramme de cas d’utilisation relatif à la gestion des locataires

1. **Diagramme de cas d’utilisation relatif à la gestion des inventaires**

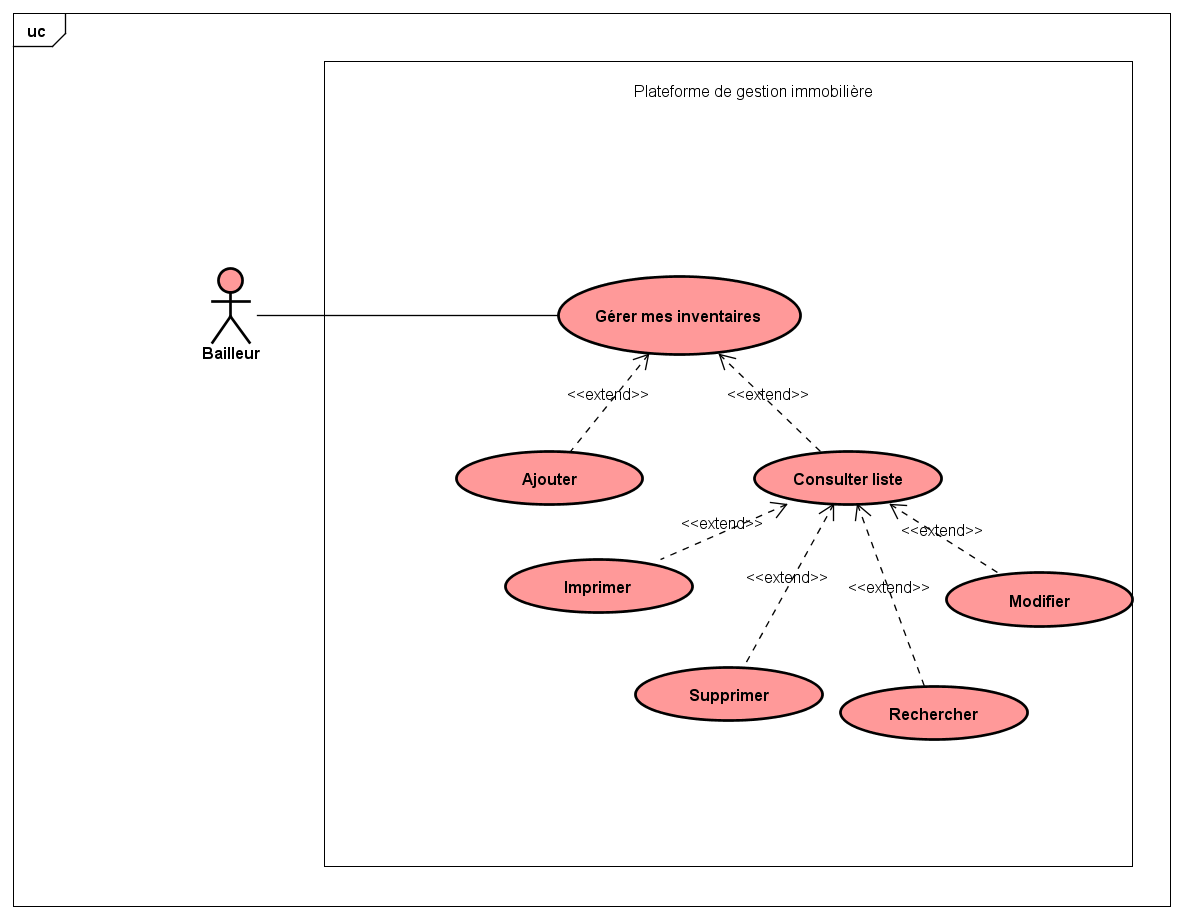


Figure 10 Diagramme de cas d’utilisation relatif à la gestion des inventaires

1. **Diagramme de cas d’utilisation relatif**

**à la consultation des factures et des quittances**

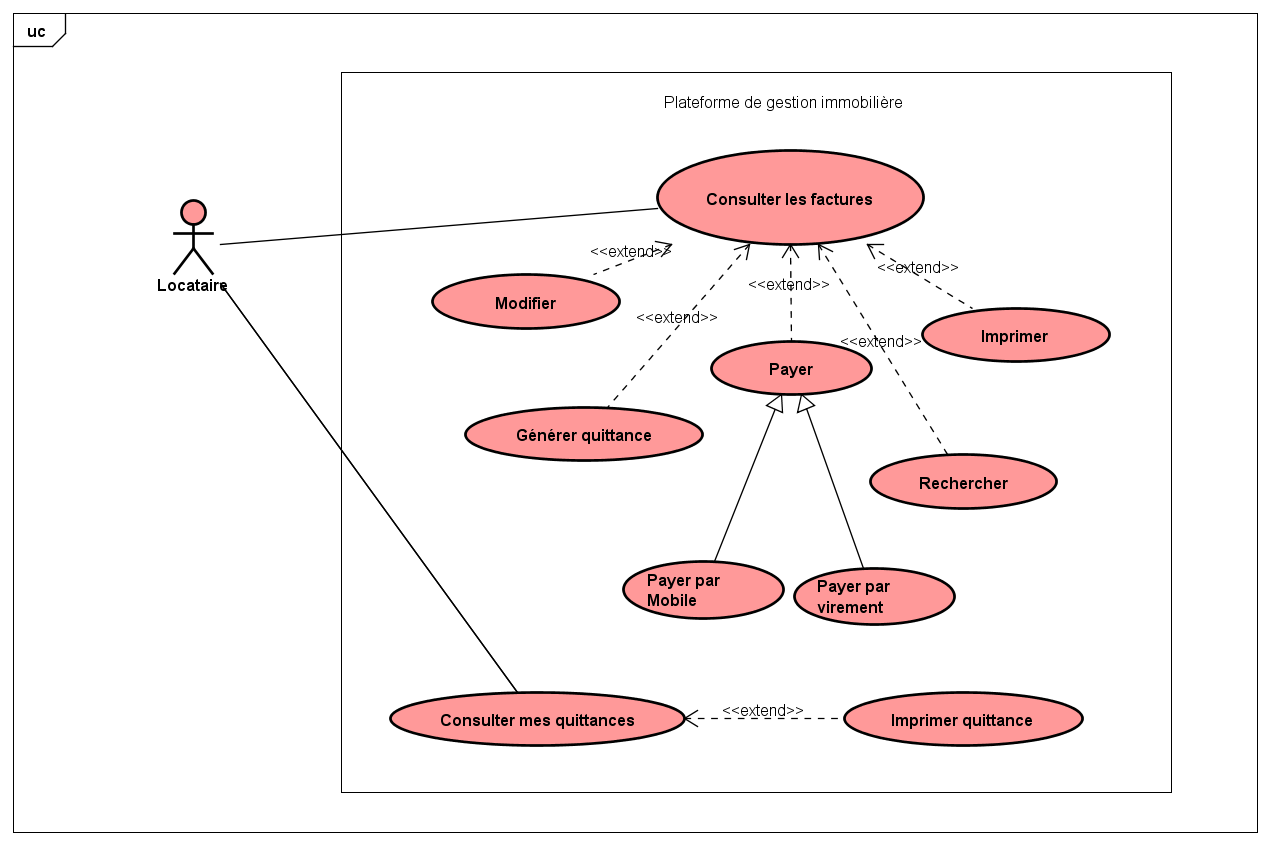


Figure 11 Diagramme de cas d’utilisation relatif à la consultation des factures et des quittances

1. **Diagramme de cas d’utilisation relatif à la gestion des états des lieux**

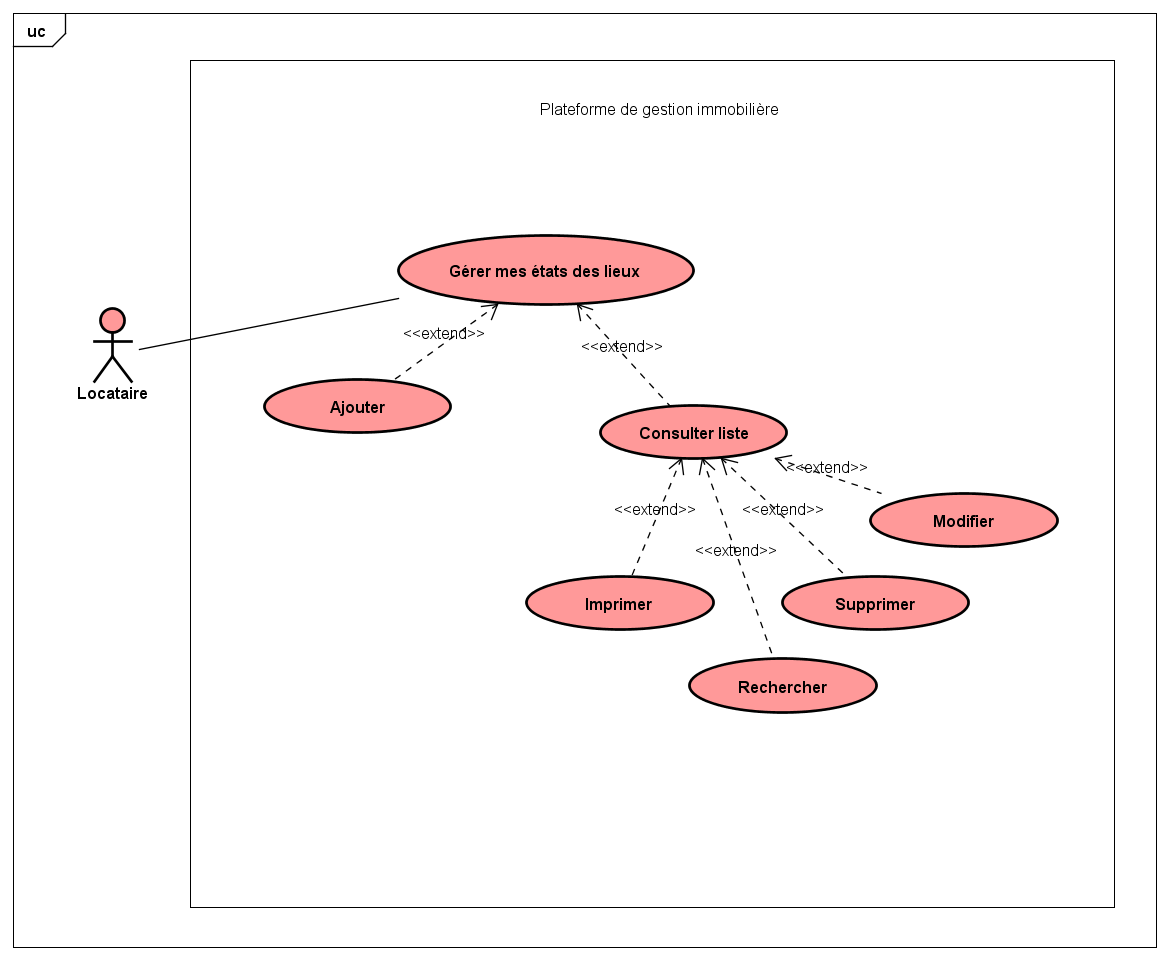


Figure 12 Diagramme de cas d’utilisation relatif à la consultation des factures et quittances du locataire

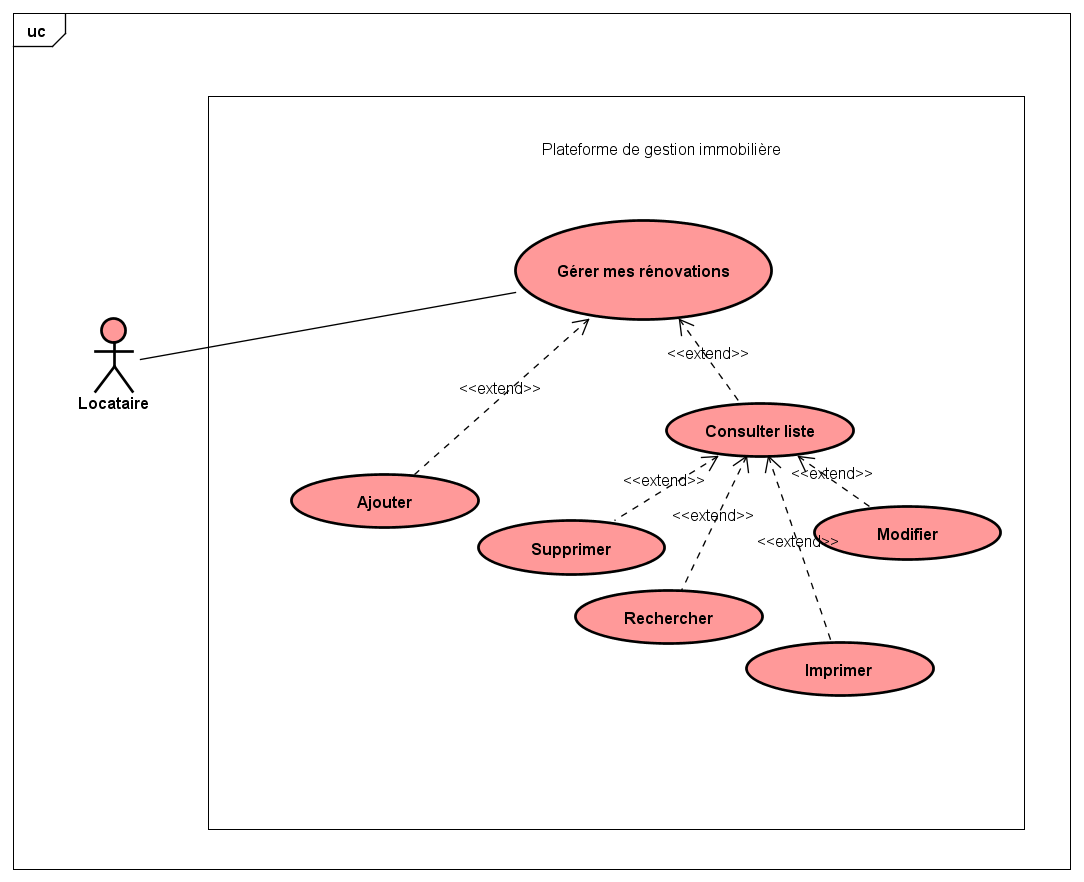
1. **Diagramme de cas d’utilisation relatif à la gestion des rénovations**

Figure 13 Diagramme de cas d’utilisation relatif à la gestion des rénovations

**VI-2 Descriptions textuelles de quelques cas d’utilisation**

Chaque cas d'utilisation, est associé à une série d'actions représentant la fonctionnalité voulue, ainsi que les stratégies à utiliser dans l'alternative où la validation échoue, ou des erreurs se produisent. Un cas d’utilisation est donc une abstraction de plusieurs chemins d’exécution. Une instance de cas d’utilisation est appelée : « scénario ». C’est un chemin particulier pris lors de l’exécution d’un cas d’utilisation. Le nombre d’instances pour un cas d’utilisation peut être très important, voire infini.

La description d’un cas d’utilisation se fait par des scenarii qui définissent la suite logique des interactions qui constitue ce cas. Cette description se fait de manière simple, par un texte compréhensible par les personnes du domaine de l’application. Elle précise également ce que fait l’acteur et ce que fait le système. Les scénarios (scénarii) peuvent être classés en :

* + - **Scénario principal ou nominal :** Il correspond à l’instance principale du cas d’utilisation. C’est le scénario typique de succès ;
    - **Scénarios secondaires :** Il peut être un cas alternatif (un choix), un cas exceptionnel ou une erreur.

La description textuelle est constituée des éléments suivants :

* **Le présupposé :** Ce sont les actions qui doivent être remplies avant le lancement du cas d’utilisation ;
* **La précondition :** C’est l’ensemble des conditions qui doivent être remplies avant le lancement du cas d’utilisation ;
* **Le déclencheur :** C’est l’élément qui déclenche le cas d’utilisation ;
* **Le dialogue :** C’est le scénario principal et le Scénario secondaire ;
* **La post condition** : Ce sont les différents mécanismes pouvant conduire à la fin de l’exécution du cas d’utilisation ;
* **L’arrêt :** C’est l’état du système à l’arrêt du cas d’utilisation. Nous présenterons dans les lignes qui suivent, les descriptions textuelles de quelques cas d’utilisation

1. Cas d’utilisation « M’authentifier »

Tableau 14 Description textuelle du cas d'utilisation "m'authentifier"

|  |  |  |  |
| --- | --- | --- | --- |
| **Cas d’utilisation m’authentifier** | | | |
| **Acteur** | | Utilisateur | |
| **Présupposé** | | Le système fonctionne correctement | |
| **Précondition** | | L’utilisateur lance l’application | |
| **Déclencheur** | | Clic sur le lien « Se connecter » | |
| **Dialogue** | | **Scénario nominal**   1. L’utilisateur clique sur « se connecter » 2. Le système affiche la page de connexion 3. L’utilisateur saisit le login et le mot de passe 4. Le système vérifie l’existence des données   5. Le système affiche la page d’accueil.  **Scénario Alternatif**   1. Si le mot de passe ou le login entré par l’utilisateur n’est pas reconnu, le scénario reprend à l’étape 2 du scénario nominal. 2. S’il y a eu une erreur interne dans le système lors du traitement de la requête, l’acteur est renvoyé à l’étape 2 du scenario nominal | |
| **Post condition de succès** | | L’acteur est connecté et la page d’accueil s’affiche | |
| **Post condition d’échec** | | L’utilisateur n’a pas pu se connecter | |

1. Cas d’utilisation « m’inscrire »

Tableau 15 Description textuelle du cas d'utilisation "m'inscrire"

|  |  |
| --- | --- |
| **Cas d’utilisation m’inscrire** | |
| **Acteur** | Visiteur |
| **Présupposé** | Le système fonctionne correctement |
| **Précondition** | Il faudrait que l’utilisateur à ajouter ne soit pas déjà dans le système |
| **Déclencheur** | Clic sur le lien « S’inscrire » |
| **Dialogue** | **Scénario nominal**   1. L’acteur clique sur le lien vers la page d’inscription permettant de créer un compte utilisateur 2. Le système affiche la page d’inscription et le formulaire à remplir 3. Il saisit les informations à remplir sur le nouveau compte 4. Le système vérifie les données (si les données saisies dans le formulaire sont valides et si l’utilisateur n’existe pas déjà dans le système) 5. Le système enregistre le compte crée dans la base de données   **Scénario Alternatif**  Les informations saisies dans le formulaire sont invalides ou l’acteur qui veut être créer existe déjà dans le système, le scénario reprend à partir du point 2 du scénario nominal. |
| **Post condition de succès** | L’acteur est inscrit et la page d’accueil s’affiche |
| **Post condition d’échec** | L’acteur n’a pas pu s’inscrire |

iii. Cas d’utilisation « ajouter un bien »

Tableau 16 Description textuelle du cas d'utilisation "ajouter un bien"

|  |  |  |  |
| --- | --- | --- | --- |
| **Cas d’utilisation ajouter un bien** | | | |
| **Acteur** | | Bailleur | |
| **Présupposé** | | Le système fonctionne correctement | |
| **Précondition** | | Avoir un compte actif et s’être authentifié dans l’application | |
| **Déclencheur** | | Clic sur le lien « Ajouter un bien » | |
| **Dialogue** | | **Scénario nominal**   1. Le bailleur clique sur l’onglet permettant d’ajouter un bien 2. Le système affiche la page et le formulaire à remplir 3. L’acteur saisit toutes les données nécessaires dans le formulaire 4. Le système vérifie les données (si les données saisies dans le formulaire sont valides) 5. Le système enregistre le bien créé dans la base de données   **Scénario Alternatif**  Si à l’étape 3 le format des données entrées n’est pas correct, un message d’erreur de format est retourné à l’acteur et on retourne à l’étape 2 | |
| **Post condition de succès** | | Un message de succès est envoyé à l’acteur | |
| **Post condition d’échec** | | Un message d’échec est envoyé à l’acteur | |

iv. Cas d’utilisation « Créer un locataire »

Tableau 17 Description textuelle du cas d'utilisation "ajouter un locataire"

|  |  |
| --- | --- |
| **Cas d’utilisation créer un locataire** | |
| **Acteur** | Bailleur |
| **Présupposé** | Le système fonctionne correctement |
| **Précondition** | Avoir un compte actif et s’être authentifié dans l’application |
| **Déclencheur** | Clic sur le lien « Ajouter un locataire » |
| **Dialogue** | **Scénario nominal**  1. Le bailleur clique sur l’onglet permettant de créer un locataire  2. Le système affiche la page et le formulaire à remplir   1. Le bailleur saisit toutes les données nécessaires dans le formulaire 2. Le système vérifie les données (si les données saisies dans le formulaire sont valides) 3. Le système enregistre le locataire créé dans la base de données   **Scénario Alternatif**  Si à l’étape 3 le format des données entrées n’est pas correct, un message d’erreur de format est retourné à l’acteur et on retourne à l’étape 2  Si à l’étape 1, si le locataire a déjà un compte l’acteur aura juste la possibilité de recherche son nom et l’inviter. |
| **Post condition de succès** | Un message de succès est envoyé à l’acteur |
| **Post condition d’échec** | Un message d’échec est envoyé à l’acteur |

v. Cas d’utilisation « Ajouter une location »

Tableau 18 Tableau 18 Description textuelle du cas d'utilisation "ajouter une location"

|  |  |
| --- | --- |
|  | **Cas d’utilisation ajouter une location** |
| **Acteur** | Bailleur |
| **Présupposé** | Le système fonctionne correctement |
| **Précondition** | Avoir un compte actif et s’être authentifié dans l’application |
| **Déclencheur** | Clic sur le lien « Ajouter une location » |
| **Dialogue** | **Scénario nominal**  1. Le bailleur clique sur l’onglet permettant d’ajouter une location  2. Le système affiche la page et le formulaire à remplir   1. Le bailleur saisit toutes les données nécessaires dans le formulaire 2. Le système vérifie les données (si les données saisies dans le formulaire sont valides) 3. Le système enregistre la location créée dans la base de données   **Scénario Alternatif**  Si à l’étape 3 le format des données entrées n’est pas correct, un message d’erreur de format est retourné à l’acteur et on retourne à l’étape 1 |
| **Post condition de succès** | Un message de succès est envoyé à l’acteur |
| **Post condition d’échec** | Un message d’échec est envoyé à l’acteur |

**VI-3 Diagramme de séquence**

1. **Présentation**

Les diagrammes des séquences permettent de décrire COMMENT les éléments du système interagissent entre eux et avec les acteurs. Ils décrivent le déroulement de chaque cas d’utilisation, en montrant la façon dont les diverses entités mises en œuvre dans le cas d’utilisation interagissent et collaborent afin de réaliser les fonctionnalités attendues. Le diagramme des séquences énumère des objets horizontalement, et le temps verticalement. Il modélise l’exécution des différents messages en fonction du temps.

1. **Formalisme d’un diagramme de séquence**

Le tableau suivant présente les éléments constitutifs d’un diagramme de séquence.

|  |  |  |
| --- | --- | --- |
| **Elément du diagramme** | **Description** | **Représentation graphique** |
| **Objet** | Ce sont des instances de classe | Objet\_1 |
| **Acteur** | Toute entité qui interagit avec le système | Acteur |
| **Ligne de vie** | Ligne verticale qui identifie l’existence de l’objet par rapport au temps |  |
| **Activation** | Boite rectangulaire sur une ligne de vie marquant la portée d’une action |  |
| **Message** | Flèche horizontale portant un texte qui indiquent les communications objet-acteur et objet-objet | Message\_1 |

Tableau 19 Formalisme du diagramme de séquence

1. **Représentation graphique des digrammes de séquences de quelques cas d’utilisation du système**
2. **Cas d’utilisation « M’authentifier »**

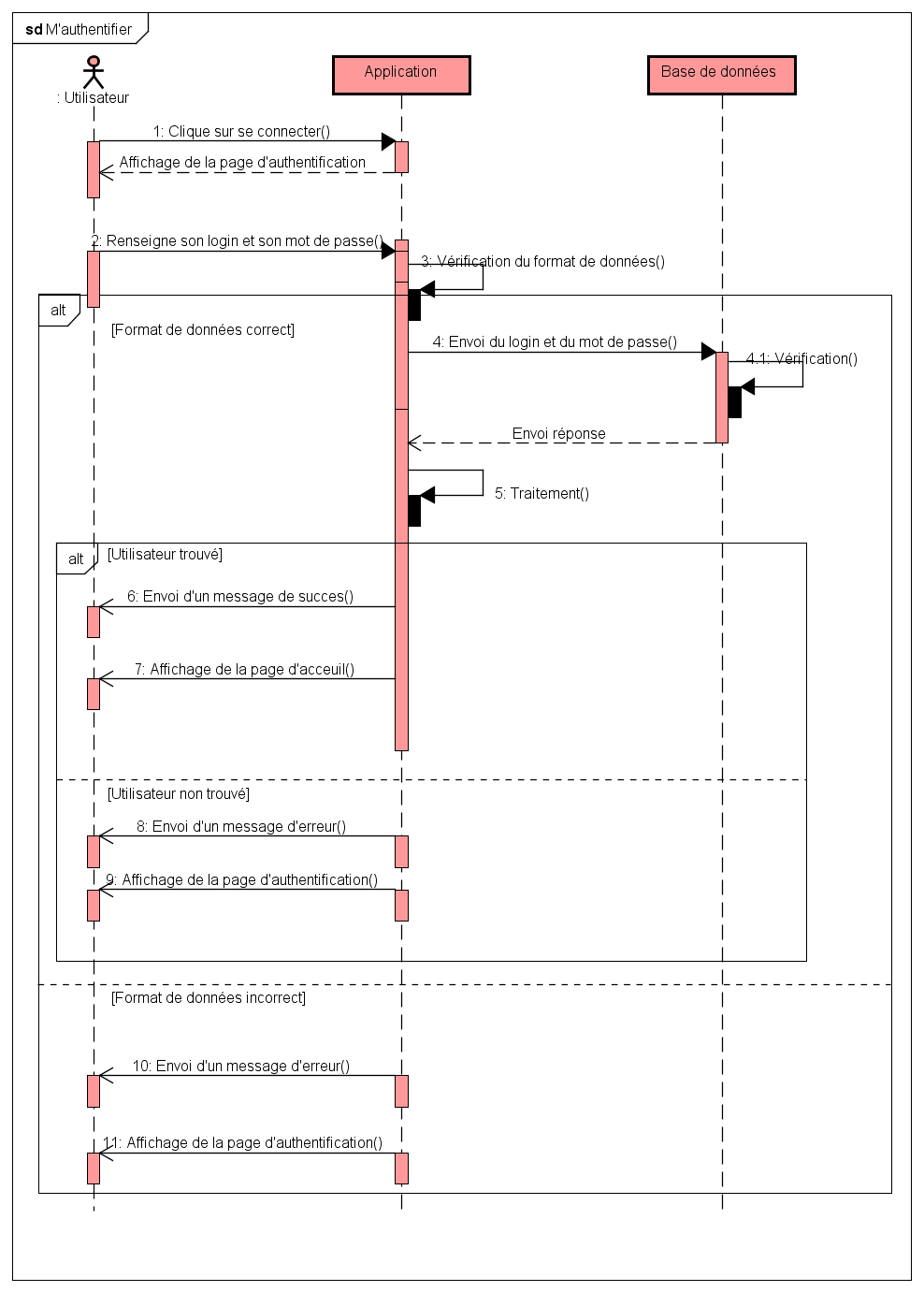


Figure 14 Diagramme de séquence du cas d'utilisation "M'authentifier"

1. **Cas d’utilisation « Ajouter un bien »**

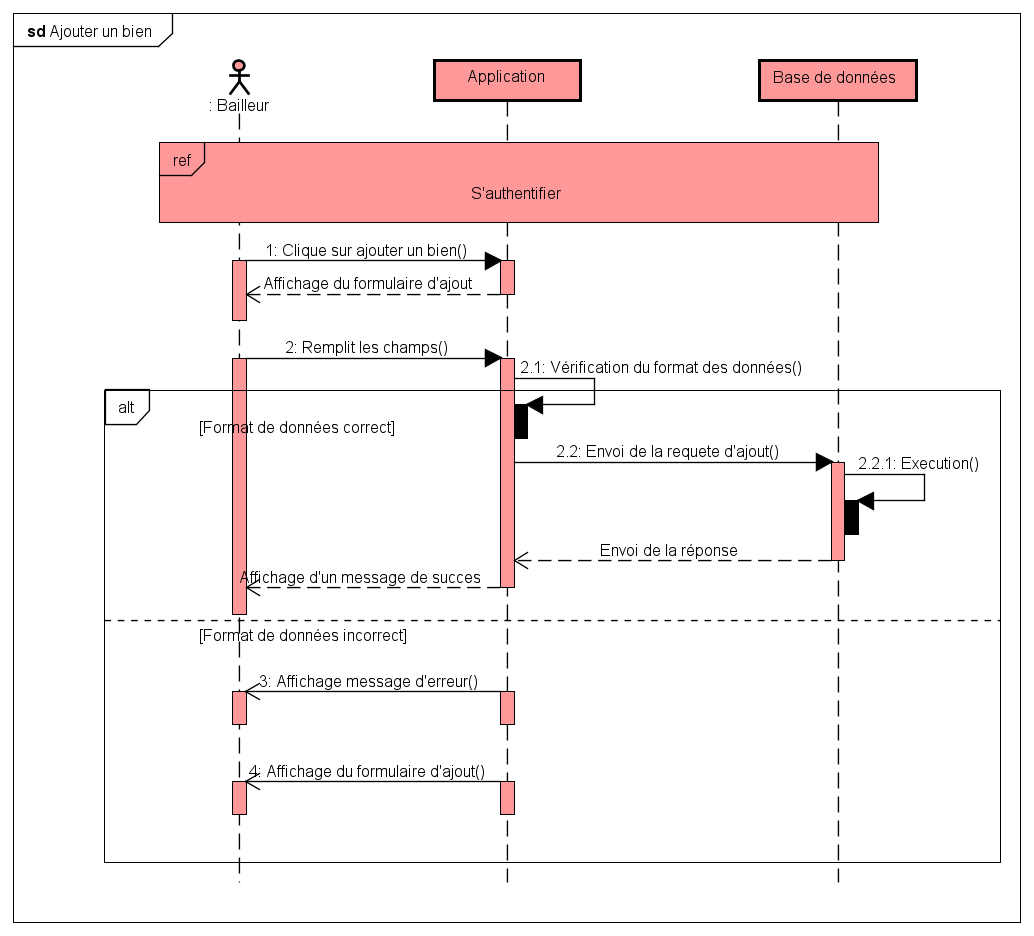


Figure 15 Diagramme de séquence du cas d'utilisation "Ajouter un bien"

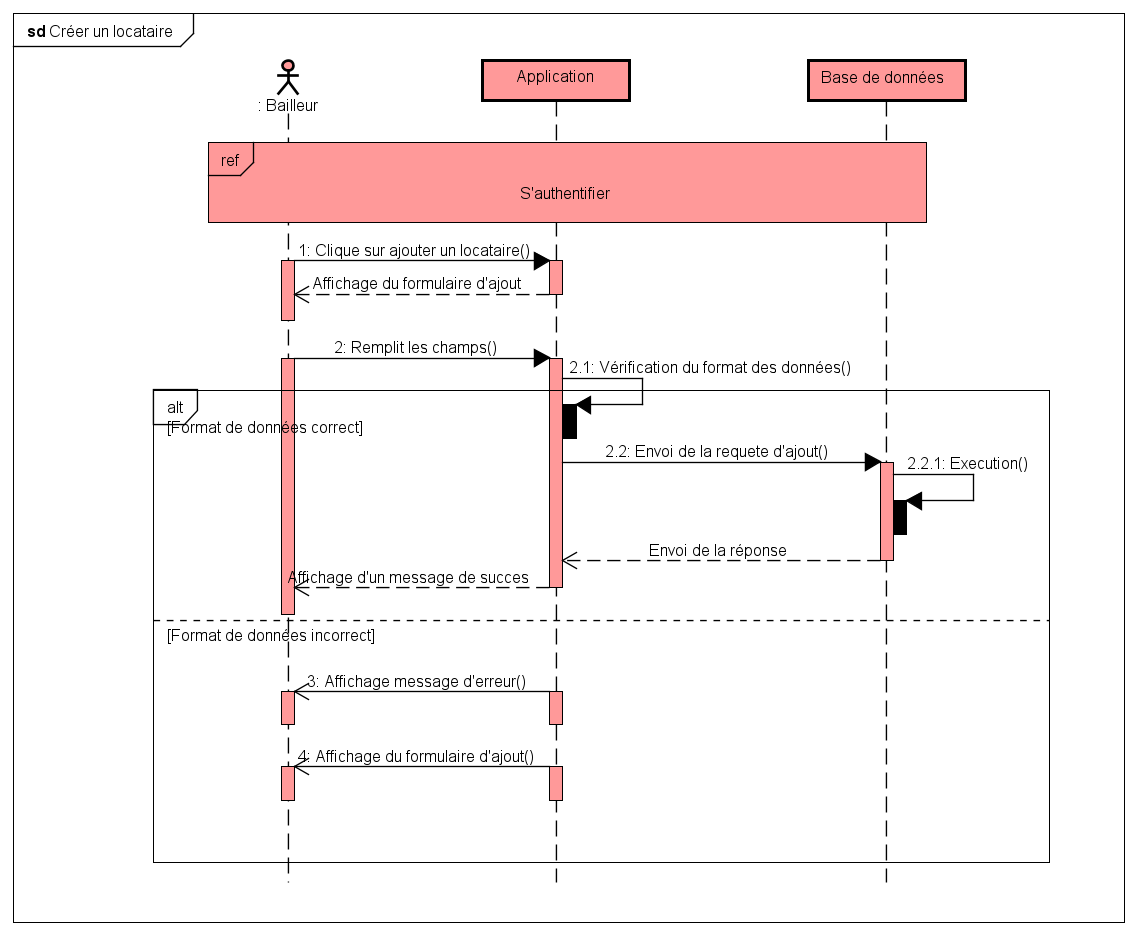
1. **Cas d’utilisation « Créer un locataire »**

Figure 16 Diagramme de séquence du cas d'utilisation "Créer locataire"

**VI-4 Diagramme d’activité**

1. **Présentation**

Les diagrammes d’activités sont les diagrammes UML utilisés pour documenter le déroulement des opérations dans un système. L’usage de ces diagrammes permet de faire apparaitre des flots de traitement induits par des processus internes par rapport aux évènements externes.

1. **Formalisme du diagramme de classe**

Tableau 20 Formalisme du diagramme d'activité

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Elément du diagramme Description** | | | | **Représentation graphique** |
| **L’activité** | | L’état d'activité marque une action faite par un objet. Il est représenté par un rectangle aux coins arrondis. | |  |
| **La transition** | | Quand un état d'activité est accompli, le traitement passe à un autre état d'activité. Les transitions (modélisées par des flèches.) sont utilisées pour marquer ce passage. | |  |
| **Les couloirs** | | Le diagramme d'activités fait intervenir les acteurs de chaque activité. Chaque activité sera placée dans une colonne (couloir) qui correspond à l'acteur. Les objets sont énumérés au-dessus de la colonne, et les barres verticales séparent les colonnes pour former les couloirs d’activités. | | |  | | --- | | UniteOrganisation\_1 | |  | |
| **L’état initial** | | Il marque le point d'entrée la première activité. Il est représenté par un cercle plein. Il ne peut y avoir qu'un seul état initial sur un diagramme | |  |
| **L’état final** | | L'état final marque la fin du déroulement des opérations modélisées. |  |
| **Barre de Synchronisation** | | Souvent, certaines activités peuvent être faites en parallèle. Pour dédoubler le traitement "Fork", ou le reprendre quand des activités multiples ont été accomplies, des barres de synchronisation sont utilisées. |  |

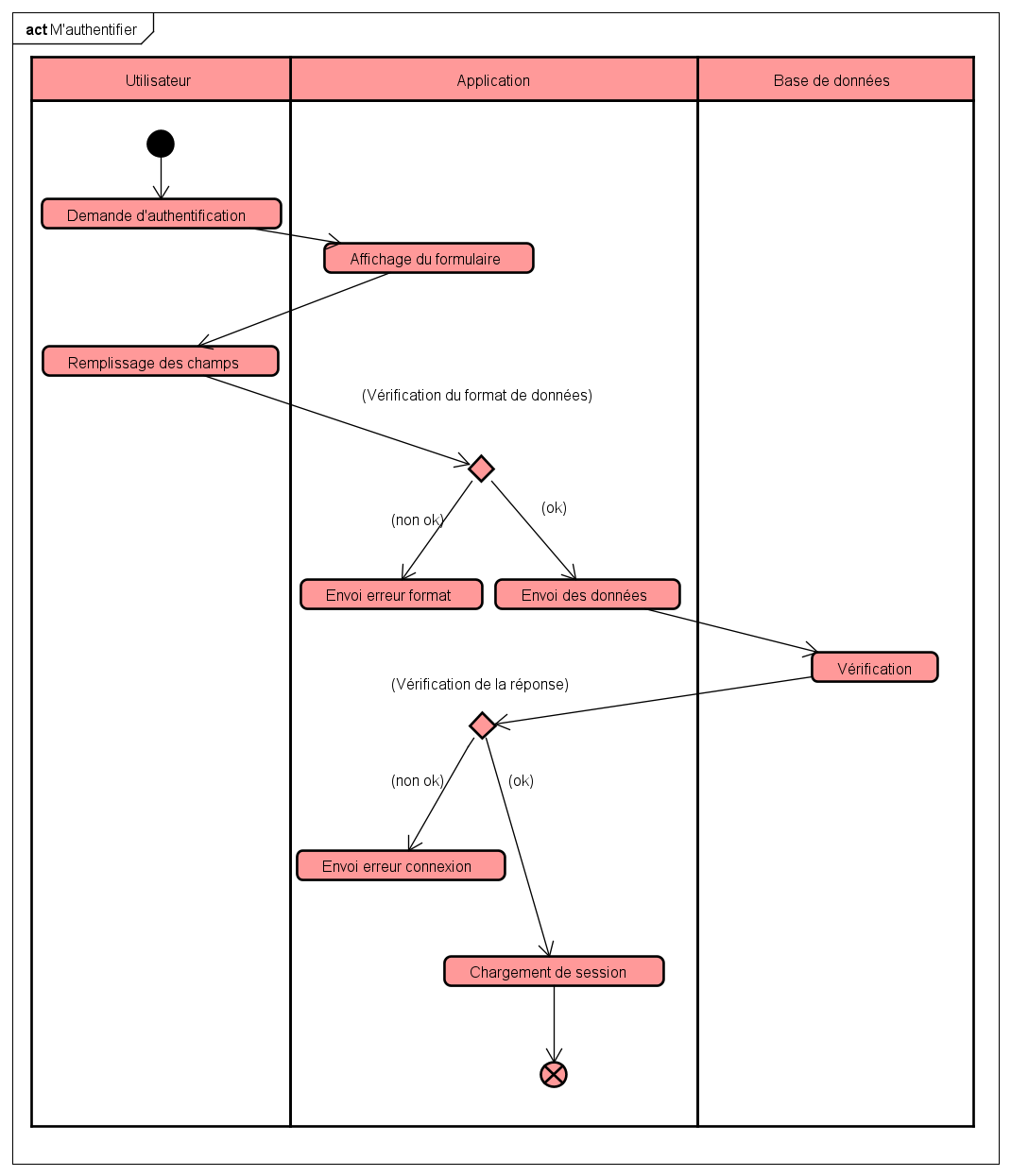
1. **Représentation graphique des digrammes d’activités de quelques cas d’utilisation du système**
2. **Cas d’utilisation « M’authentifier »**

Figure 17 Diagramme d'activité du cas d'utilisation "M'authentifier"

* 1. **Cas d’utilisation « Ajouter un bien »**

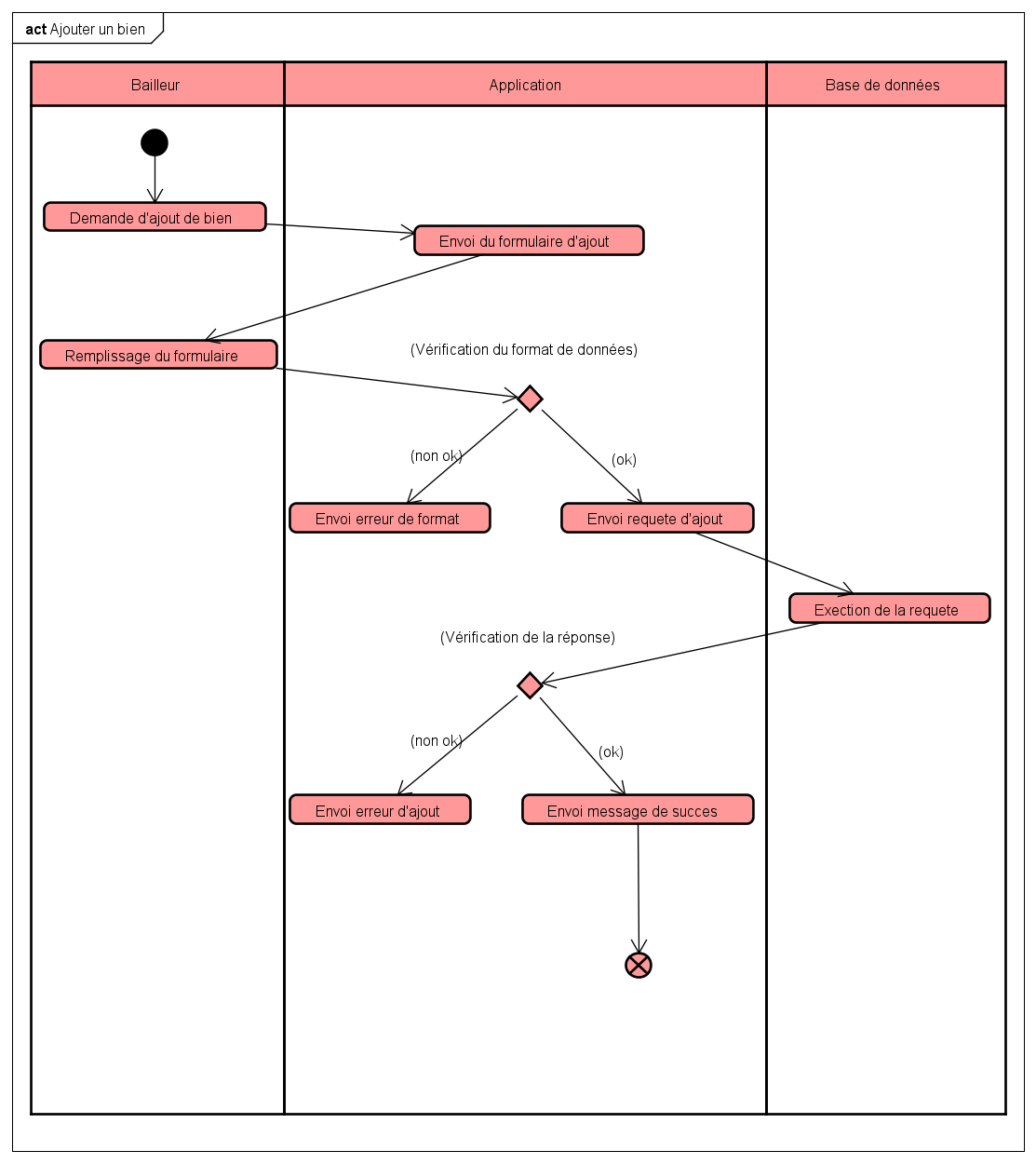


Figure 18 Diagramme d'activité du cas "Ajouter un bien"

* 1. **Cas d’utilisation « Créer un locataire »**

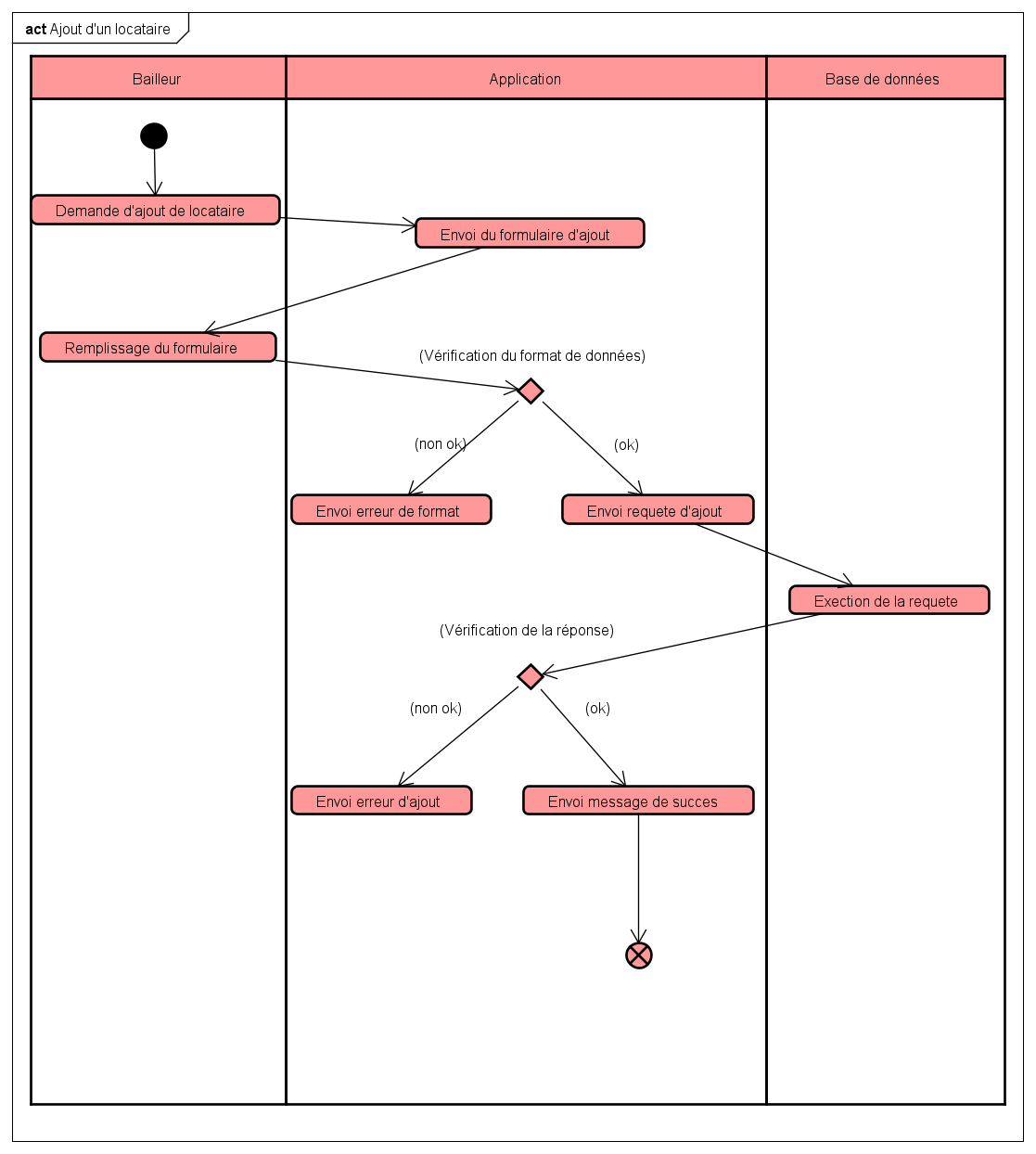


Figure 19 Diagramme d'activité du cas "Créer un locataire"

**CONCLUSION**

L’analyse du processus de Mise en place d’une plateforme de gestion immobilière nous a permis de parcourir plusieurs notions afin de produire une analyse qui répond aux exigences. Ainsi nous avons de prime-abord décrit l’existant, donné les limites de cet existant pour ensuite proposer une solution ; par la suite, nous avons parlé de la démarche à utiliser dans le cadre de notre projet, et enfin, nous avons pu étudier en détail la solution proposée en modélisant nos différents diagrammes grâce aux logiciels **ASTAH UML 8.4**. Il en ressort que dans ce dossier les éléments pris en compte nous ont permis de matérialiser les besoins fonctionnels ; ces derniers nous seront d’une importance capitale dans l’expression des besoins techniques. Le dossier d’analyse à travers les différentes activités réalisées, nous a offert des outils qui nous permettront d’amorcer de façon plus aisée le dossier de conception.

**DOSSIER DE CONCEPTION**

***Préambule***

Le dossier de conception permet de modéliser dans son ensemble la solution proposée et de recueillir les informations nécessaires à la mise sur pied d’une base de données complexe et efficace. De ce fait il prévoit le système futur.

***Aperçu***

**INTRODUCTION**

**I-OBJECTIFS DE LA CONCEPTION**

**II-PRESENTATION DES DIAGRAMMES INTERVENANTS**

**CONCLUSION**

**INTRODUCTION**

Dans la partie précédente, il était question pour nous d’analyser notre système, nous nous sommes surtout intéressés aux besoins des utilisateurs. Cela dit, après avoir analysé notre système, nous allons passer à la conception de notre futur système. Maintenant nous réfléchirons à l’architecture du système qui permettra de répondre à tous ces besoins. Nous présenterons donc par la suite des vues qui montrent la structure statique du système, et l’architecture logicielle de celui-ci.

**I-OBJECTIFS DE LA CONCEPTION**

L’objectif de cet exercice est de présenter la structure du futur logiciel. De ce fait, le dossier de conception que nous allons élaborer va permettre de :

* Dégager les grandes fonctionnalités du système ;
* Spécifier chaque module dans le but de faciliter la réalisation du système ;
* Constituer le manuel de référence qui donne la description détaillée de chaque unité de programme (écran, état) et de contrôle exigé ;
* Préparer le passage à l’étude technique.

A cet effet, notre dossier de conception sera structuré autour du diagramme de classe.

**II-PRESENTATION DES DIAGRAMMES INTERVENANTS**

**II-1 Diagramme de classe**

1. **Présentation**

Le diagramme de classe exprime la structure statique du système en termes de classe et de relation entre ces classes. Autrement dit, ce diagramme permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d’utilisation. Il est donc important de noter qu’un même objet peut très bien intervenir dans la réalisation de plusieurs cas d’utilisation. L’intérêt du diagramme de classe est de modéliser les entités du SI.

1. **Formalisme**

Le diagramme de classe met en évidence d’éventuelles relations entre ces classes. Les principaux éléments du diagramme de classe sont : **les classes, les associations, généralisation (héritage), la composition, l’agrégation et la multiplicité**.

Tableau 21 Formalisme du diagramme de classe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Eléments du diagramme** | **Description** | **Représentation** | | |
| **La classe** | Est un ensemble d’attributs et de méthodes regroupés au sein d’une même entité |  | | |
|  | Classe\_1 |  |
| -  -  -  Attribut\_1  Attribut\_2  Attribut\_3  :  int  :  int  :  int |
|  |
| **L’association** | C’est la relation générique entre deux classes | 0..1  0..\* | | |
| **La multiplicité** | Définit les nombres de fois minimum et maximum qu’une classe peut être associée à une autre. Il y en plusieurs | * 0..1 * 0.. \* * 1..1 * 1..\* | | |
| **La généralisation** | Traduit le concept d’héritage. Est utilisée pour faire le lien entre un (ou plusieurs) objet(s) spécialisé(s) et l’objet générique dont il(s) dérive(nt) |  | | |
| **L’agrégation** | Est une relation d’appartenance entre des classes, où les classes peuvent exister indépendamment (la destruction de l’agrégat n’entraine pas forcement  celle de l’agrégé) | 0..1  0..\* | | |
| **La composition** | Est un cas particulier d’agrégation. Ici, les classes composantes ne peuvent  exister sans leur composite | 0..1  0..\* | | |

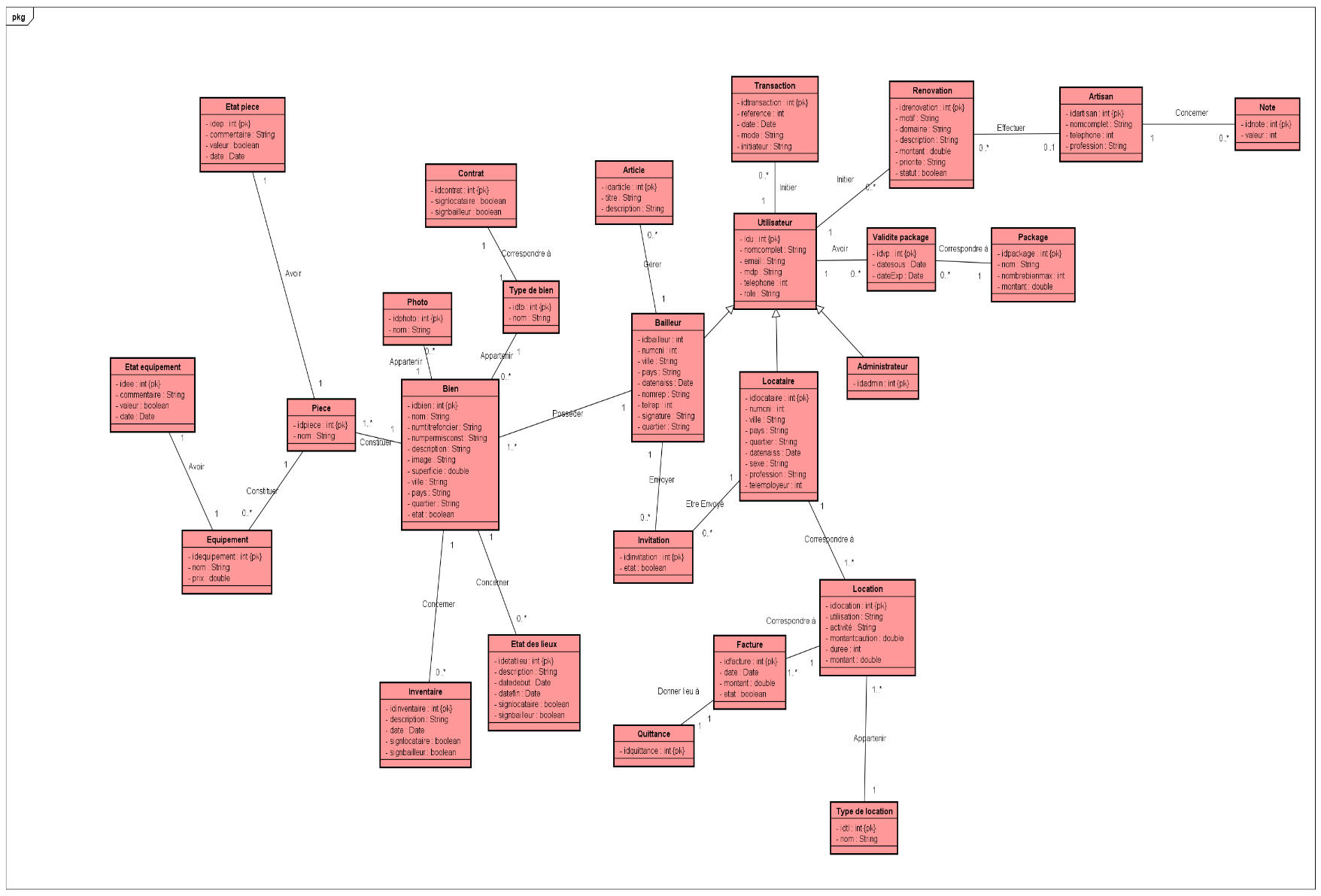
1. **Diagramme de classe du système**

Figure 20 Diagramme de classe du système

**CONCLUSION**

Nous avons pu au terme de cette partie ressortir de manière détaillée les différentes parties de la conception de notre projet au moyen des objectifs fixés et du diagramme de classes. A travers cette étape, nous avons pu avoir une vision générale sur l’application qui sera réalisée dans le dossier de réalisation.

**DOSSIER DE REALISATION**

***Préambule***

Dans le dossier de réalisation, nous présentons les différents outils nécessaires à la mise en œuvre de la solution. Il sera donc question dans cette section de présenter les technologies utilisées et les diagrammes intervenants

***Aperçu***

**INTRODUCTION**

**I-EQUIPE DE DEVELOPPEMENT ET TECHNOLOGIES CHOISIES**

**II-PHASE D’IMPLEMENTATION**

**III-PHASE DE DEPLOIEMENT**

**CONCLUSION**

**INTRODUCTION**

La phase de réalisation est la partie qui traite de la mise en œuvre de la solution. Elle décrit l’application en tant qu’instruction visible et manipulable et représente une conclusion de toutes les autres parties présentées ci-dessus à savoir le dossier d’analyse et le dossier de conception. C’est donc une concrétisation de la solution proposée. Pour ce faire, cette partie s’étalera sur les points suivants : le choix des outils, la phase d’implémentation et la phase de déploiement.

**I-EQUIPE DE DEVELOPPEMENT ET TECHNOLOGIES UTILISEES**

**I-1 Equipe de développement**

Le tableau suivant présent les différents membres ayant participé à la réalisation de ce projet :

Tableau 22 Equipe de développement du projet

|  |  |  |
| --- | --- | --- |
| **Intervenant** | **Fonction** | **Rôle** |
| **Mme NANA Patience** | Enseignante à l’IAI Cameroun | Encadrant académique |
| **M. Luc Victoire KANKO** | Directeur Technique de INTERNIS | Encadrant professionnel |
| **M. FEUMBA NJIKI Alexandre** | Directeur Général de INTERNIS | Maitre d’ouvrage |
| **M. KANA MBOMA Wilfried Hariel** | Etudiant en Informatique | Analyste-concepteur, développeur |

**I-2 Ressources matérielles**

Tableau 23 Ressources matérielles du projet

|  |  |
| --- | --- |
| **Outils** | **Caractéristique** |
| **Laptop** | * Marque : Lenovo * Processeur Intel Core i5 CPU * HDD 500Go * RAM 8Go |
| **Clé USB** | * Marque : Sandisk * Capacité : 32Go |
| **Modem Camtel** | Bande passante 4Mo/s |

**I-3 Ressources logicielles**

Pour la mise en œuvre de la solution, nous avons besoins des logiciels suivants :

Tableau 24 Ressources logicielles du projet

|  |  |  |  |
| --- | --- | --- | --- |
| **Logiciel** | | **Rôle** | |
| **MYSQL 5.7.33** | | Serveur de base de données | |
| **GOOGLE CHROME** | | Navigateur web | |
| **MS WINDOWS 10** | | Système d’exploitation | |
| **ASTAH UML 8.4** | | Un atelier de génie logiciel permettant de modéliser les diagrammes du système. | |
| **GIT 2.24.0.windows.2** | | Outil de versionnage d‘application | |

**I-4 Langages de programmation et Framework**

Pour la réalisation des différentes applications, nous avons utilisé les technologies les plus récentes et accessibles, ceci pour faciliter la maintenance évolutive des programmes.

Tableau 25 Langages et Framework utilisés dans le projet

|  |  |
| --- | --- |
| **Langage et Framework** | **Rôles** |
| **HTML 5** | Langages utilisés pour créer et mettre en forme des pages web |
| **CSS 3** |
| **JavaScript** | Langage de programmation permettant de créer notre serveur et notre client |
| **LARAVEL 8** | Framework PHP permettant créer le backend de notre application |
| **SQL** | Langage utilisé pour communiquer avec la base de données. |

**II-PHASE D’IMPLEMENTATION**

**II-1 Aspect sécurité de l’application**

1. **Sécurisation du mot de passe**

Dans la base de données, les mots de passe sont hachés, par conséquent illisibles par tout utilisateur, même les administrateurs du système.

1. **Sécurité de la connexion**

Chaque utilisateur possède un espace de travail, administrateur comme utilisateur simple. Les utilisateurs doivent s’authentifier en entrant leurs login et mot de passe avec l’usage des tokens JWT pour renforcer la sécurité de notre système. L’administrateur a tous les privilèges et pourra activer ou bloquer un compte utilisateur comme il le souhaite.

**II-2 Architecture logicielle (design pattern MVC)**

Le **Modèle-vue-contrôleur** ou **MVC** est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes

* La partie Modèle
* La partie Vue
* La partie Contrôleur.

**La partie Modèle** d’une architecture MVC encapsule le logique métier (business logic) ainsi que l’accès aux données. Il peut s’agir d’un ensemble de fonctions (Modèle procédural) ou de classes (Modèle orienté objet).

**La partie Vue** s’occupe des interactions avec l’utilisateur : présentation, saisie et validation des données. Ici c’est notre application cliente

**La partie Contrôleur** gère la dynamique de l’application. Elle fait le lien entre l’utilisateur et le reste de l’application.

Ce motif est utilisé par de nombreux Framework pour applications web tels que Laravel, Symphony et plusieurs autres. La figure ci-dessous nous permet de mieux comprendre le concept du MVC.

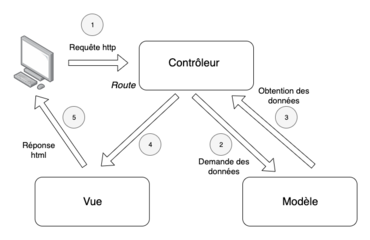


Figure 21 Représentation du MVC

Source : https://luckygg.tistory.com/182

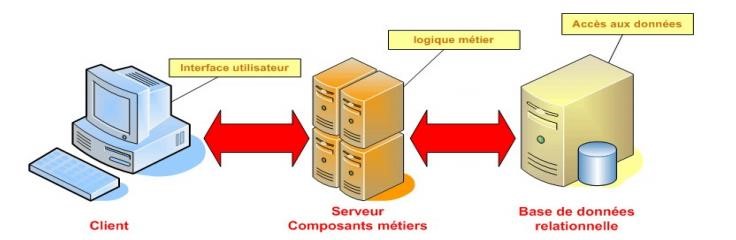
**III-PHASE DE DEPLOIEMENT**

**III-1 Architecture physique de l’application**

L’architecture physique de l’application définit la manière dont les composants physiques utiles à l’application sont liés entre eux. Nous avons parmi les nombreuses architectures logicielles qui existent, choisi l’architecture n-Tiers.

L'architecture n tiers, est une architecture client-serveur dans laquelle coexistent et sont maintenus des modules indépendants permettant le rendu d'une interface utilisateur (GUI), les processus logiques, fonctionnels et métiers ainsi que l'accès aux données. On parle donc ici d'une infrastructure physique qui va servir de support à une infrastructure logicielle (l'infrastructure trois tiers sous-tend l'infrastructure logicielle). En effet, n'importe quelle application peut être découpée en trois parties : une partie interface graphique, une partie fonctionnelle, et une partie de stockage de données. Et c'est à ces besoins précis que l'architecture trois-tiers s'est dessinée en découpant trois parties distinctes :

* **La couche présentation** (premier niveau) : Elle correspond à la partie de l’application visible par les utilisateurs. On parle d’Interface Homme-Machine (IHM). On conçoit facilement que cette interface peut prendre de multiples facettes sans changer la finalité de l’application. La couche présentation relaie les requêtes de l’utilisateur en destination de la couche métier, et en retour lui présente les informations renvoyées par les traitements de cette couche.
* **La couche métier** (second niveau) : Elle correspond à la partie fonctionnelle de l’application, celle qui implémente la « logique », et qui décrit les opérations que l’application opère sur les données en fonction des requêtes des utilisateurs effectués au travers de la couche présentation. Les différentes règles de gestion et de contrôle sont mises en règles dans cette couche. La couche métier offre des services applicatifs et métiers à la couche présentation. Pour obtenir ces services, elle appuie les cas échéant sur les données du système, accessible au travers de la couche inférieure.
* **La couche d’accès aux données** (troisième niveau) : Elle consiste en la partie gérant l’accès aux données du système. Ces données peuvent être propres au système, ou gérées par un autre système. La couche n’a pas à s’adapter à ces deux cas, ils sont transparents pour elle et elle accède aux données de manière uniforme.



Source : [*http://remy*](http://remy)[*-manu.no-ip.biz/Java/Tutoriels/J2EE/Introduction.html*](http://remy-manu.no-ip.biz/Java/Tutoriels/J2EE/Introduction.html)

Figure 22 Déploiement de l'architecture n-tiers

**III-2 Diagramme de déploiement**

1. **Présentation**

Le diagramme de déploiement est une vue statique qui sert à représenter l’utilisation de l’infrastructure physique par le système et la manière dont les composants du système sont repartis ainsi que les relations entre eux. Le diagramme de déploiement se rapproche de la réalité physique puisqu’il identifie les éléments matériels, leur disposition physique et la disposition de composants sur ces éléments matériels. Le diagramme de déploiement permet donc de représenter l’architecture physique d’un système.

1. **Formalisme**

Les éléments constitutifs du diagramme de déploiement sont :

Tableau 26 Formalisme du diagramme de déploiement

|  |  |  |
| --- | --- | --- |
| **Eléments du diagramme** | **Description** | **Représentation** |
| **Le composant** | Est une entité logicielle d’un système | InstanceComposant\_1 |
| **Le nœud** | Est un équipement matériel du système | Noeud\_1 |
| **La dépendance** | Est un lien entre deux composants qui traduit une dépendance |  |
| **L’association** | Est un lien de communication entre les nœuds du système |  |

1. **Diagramme de déploiement notre système**

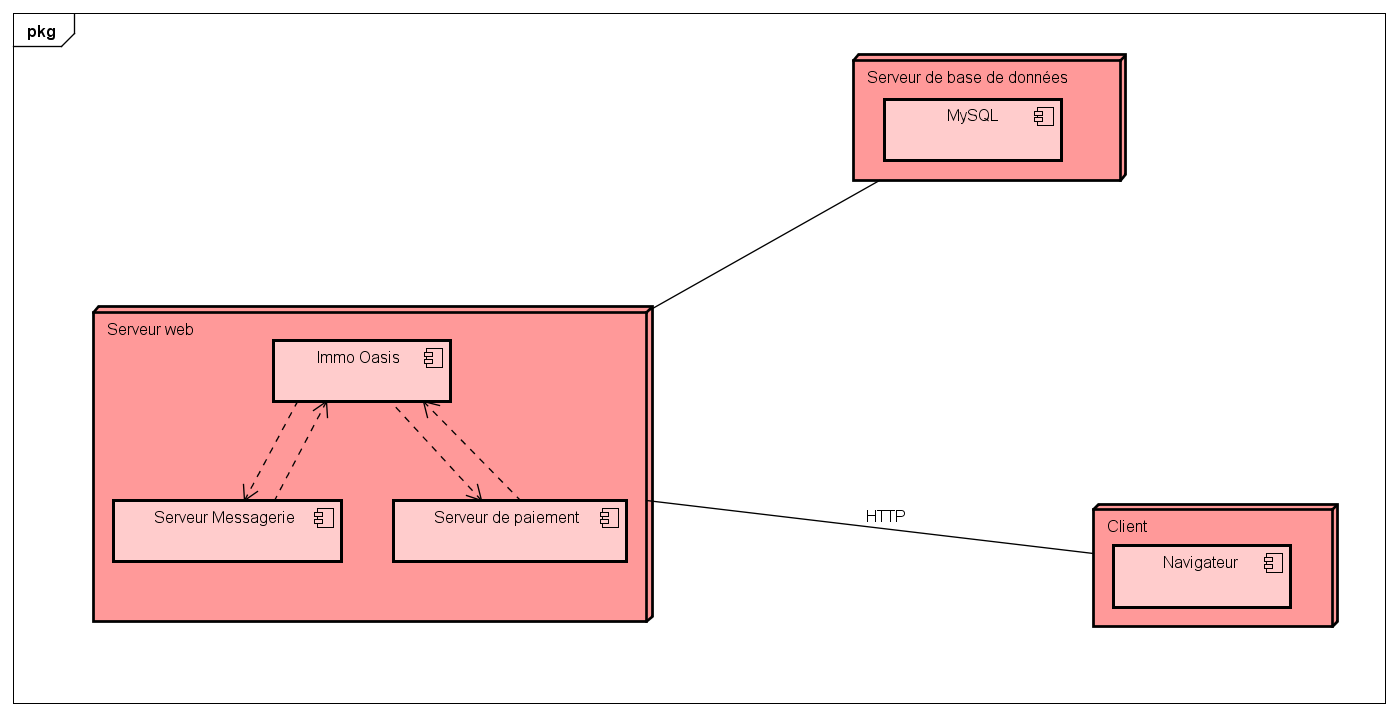


Figure 23 Diagramme de déploiement du système

**CONCLUSION**

Au moyen de langages de programmation et de différentes technologies, nous avons pu implémenter la solution par nous proposée. Pour profiter de tous les atouts de notre application, il faut faire recours à un guide d’utilisation. Pour cela, nous consacrerons la dernière partie de notre travail à présenter comment s’utilise notre application.