

RAPPORT FINAL DE PROJET

Titre du projet : *LOGICIEL DE GESTION ET DE
FACTURATION LOCATIVE*

Durée du Projet : 4 semaines

Domaine d'intervention : Social/ administration des biens d'un
particulier

Zone d'intervention : Echelle Local

Partenaires : 19M2412 TCHOUGWEN TEWO Brice Jordan
18T2482 NGUEGANG FEUMOE Ouvane Auxilia
18T2485 FOKA TETANG Algor Junior
18T2897 FEUDJIO TSAGMO Simplicie Jordan
19M2381 TCHOUAMOU PAYONG Thierry Pavone

Chef projet : 19M2412 TCHOUGWEN TEWO Brice Jordan

Superviseur : MONTHE Valéry

Table des matières

INTRODUCTION.....	3
I. BESOINS ET OBJECTIFS DU PROJET.....	4
1. Besoins fonctionnels	4
2. Besoins non-fonctionnels.....	4
II. CONCEPTION.....	4
III. PLAN DE TEST.....	10
IV. DEVELOPPEMENT	11

INTRODUCTION

Dans le cadre du cours d'INF 3055, il nous est demandé de réaliser un projet de groupe. A cet effet, nous avons choisi comme projet un ***LOGICIEL DE GESTION ET DE FACTURATION LOCATIVE***, celui-ci dans le but de résoudre le problème des bailleurs. Ce projet permettra donc au bailleur de connaître le statut de chacun de ses locataires (***qui paye quand ? combien ? qui a des aérés ? de combien ?...***) de gérer efficacement les sous factures d'eau, d'électricité, d'autre part il permettra de rappeler les dates auxquelles ils doivent payer. Ce rapport tient donc lieu de résumer des différentes activités que nous avons eu à mener pour la réalisation de ce projet.

I. BESOINS ET OBJECTIFS DU PROJET

1. Besoins fonctionnels

- Les différentes propriétés du bailleur sont enregistrées dans la base de données.
- Pour chaque locataire, le bailleur possède une trace de tous ses paiements ainsi que toutes ses informations.
- Après chaque paye, le bailleur doit enregistrer la facture réglée par le client.
- Le bailleur a la possibilité d'ajouter des locataires pour des propriétés libres.
- Le bailleur a la possibilité de supprimer ceux qui ne sont plus en location chez lui.
- Les différentes factures du locataire, sont calculées à partir de celle du compteur principal.

2. Besoins non-fonctionnels

- Fiabilité : entrée correcte (syntaxiquement et fonctionnellement).
- Convivialité : la facilité d'utilisation des interfaces proposées.
- Extensibilité maintenabilité : Ajout de fonctionnalités d'analyse et de manipulation des polynômes résultats.
- Robustesse : tolérant aux pannes.

II. CONCEPTION

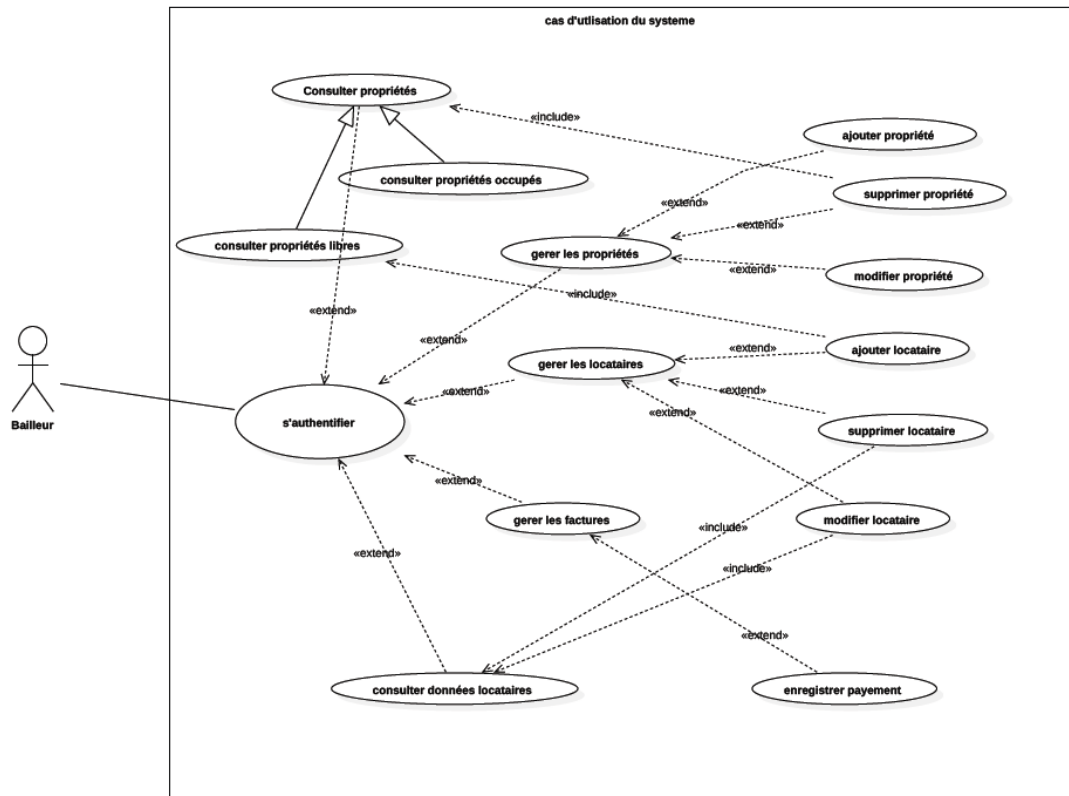
Dans le but d'apporter une solution au problème posé, nous avons définis les composants, relations entre ces composants, relations avec son environnement

- Outils et logiciels : IDE (Eclipse, NetBeans), **JRE** (Java Runtime Environment), ou **JDK** (Java Development Kit).
- Base de données : MySQL.
- Eléments physiques : ordinateurs, modem...

Pour une conception détaillée nous avons opté pour l'utilisation de certains diagrammes.

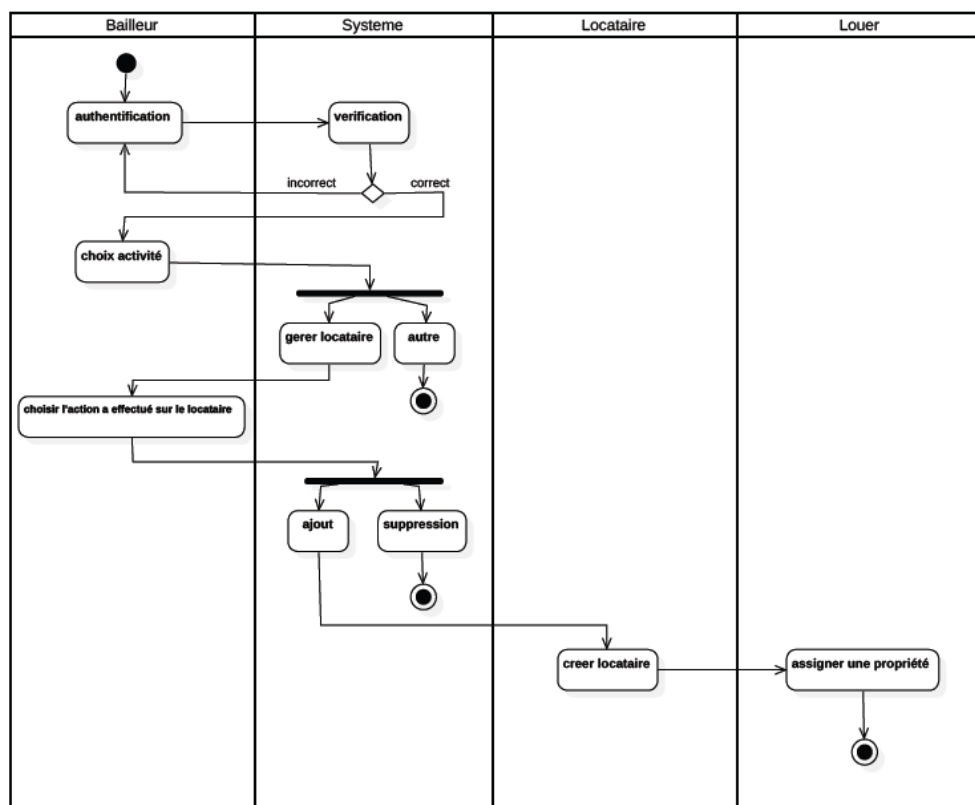
1. Diagramme de cas d'utilisation

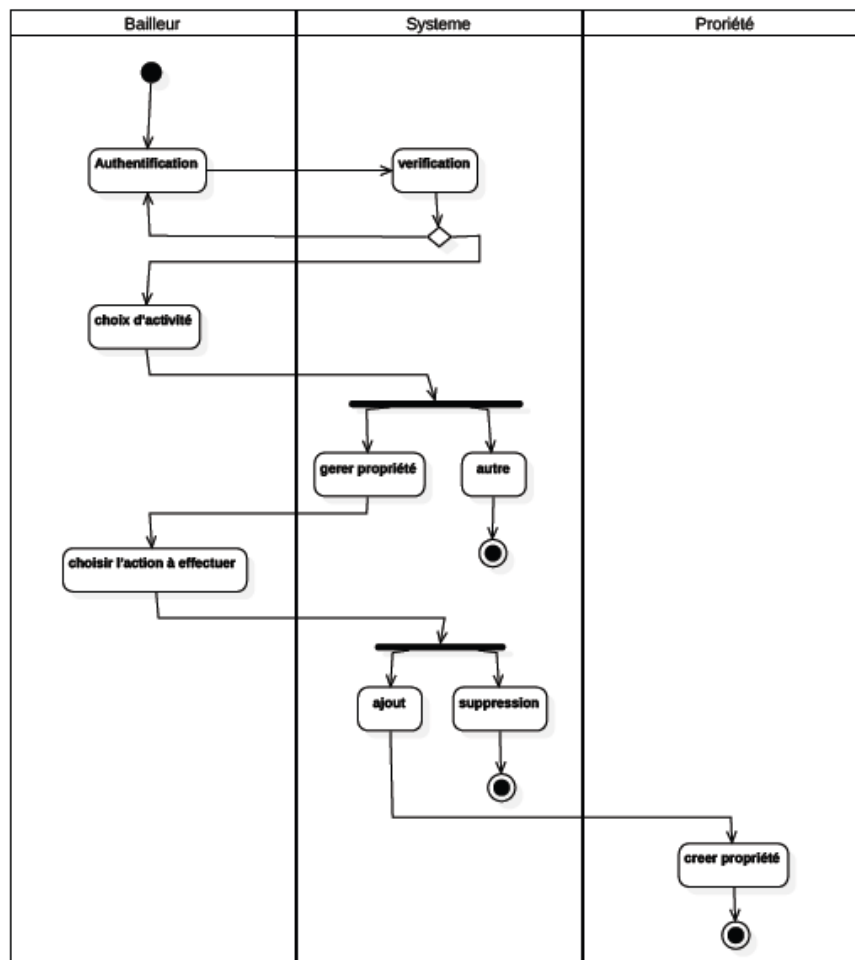
Model1::UseCaseDiagram1



2. Diagrammes d'activités

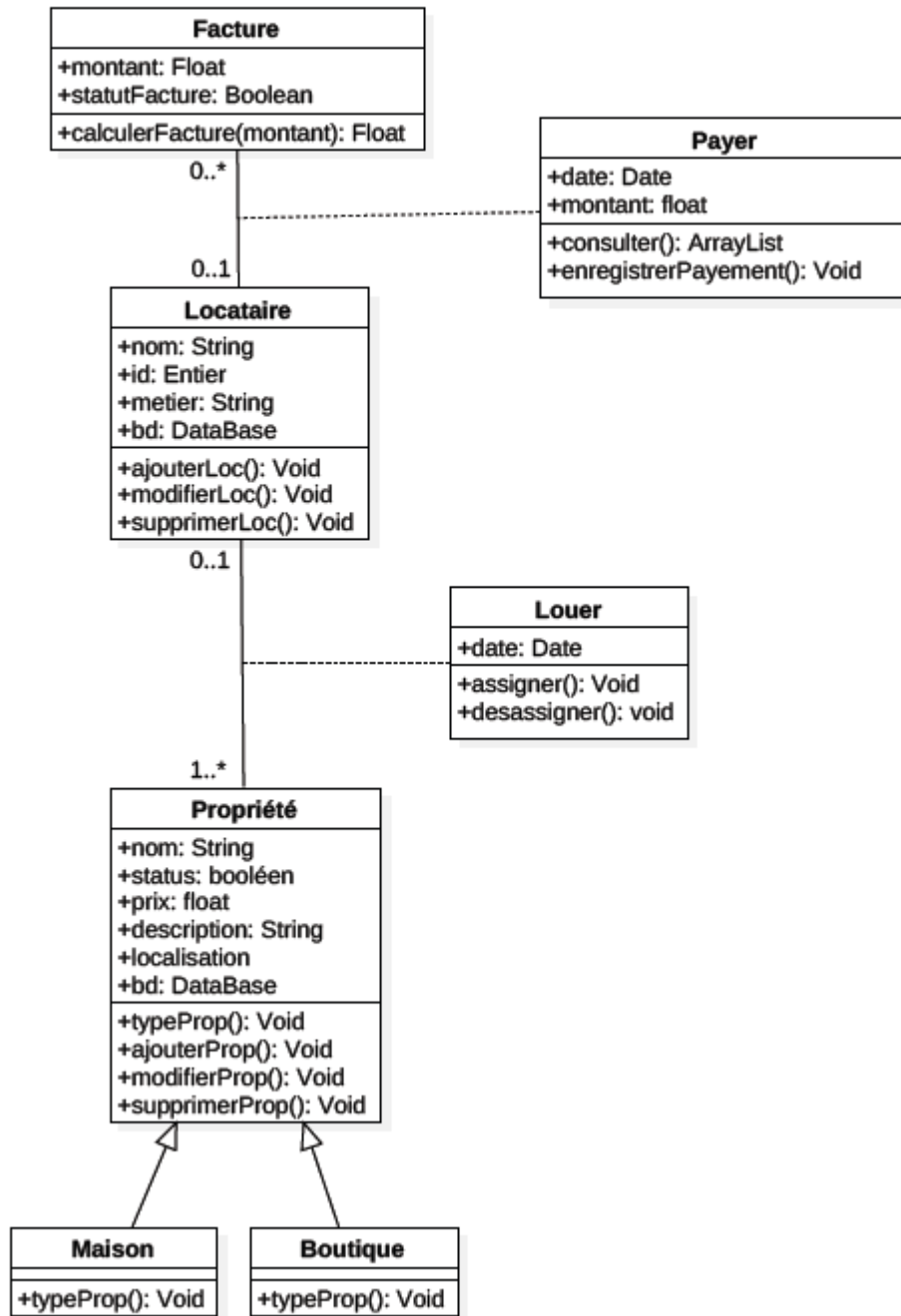
Activity1::ActivityDiagramAjoutLocataire





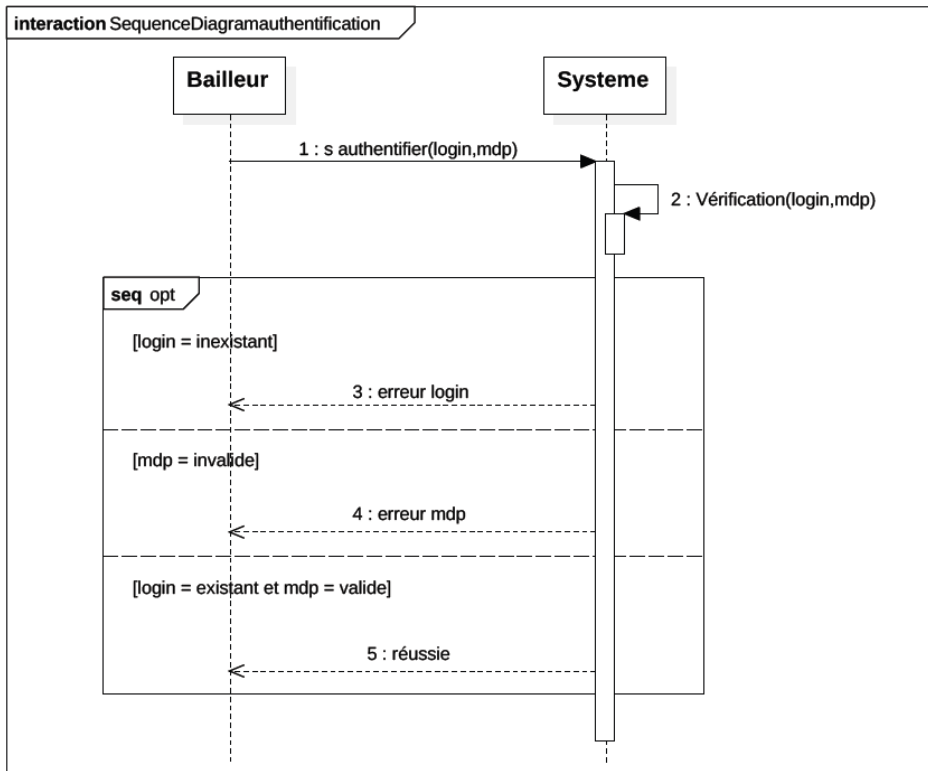
3. Diagramme de classes

Model1::ClassDiagram1

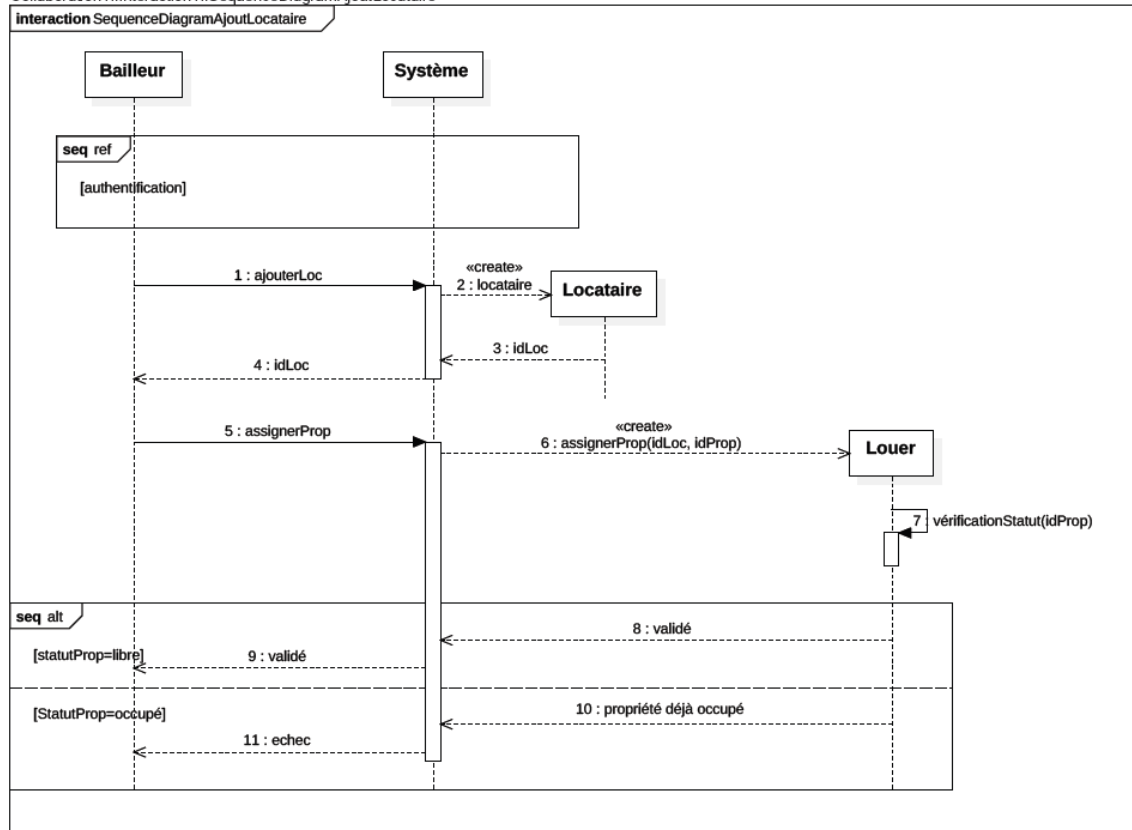


4. Diagrammes de séquences

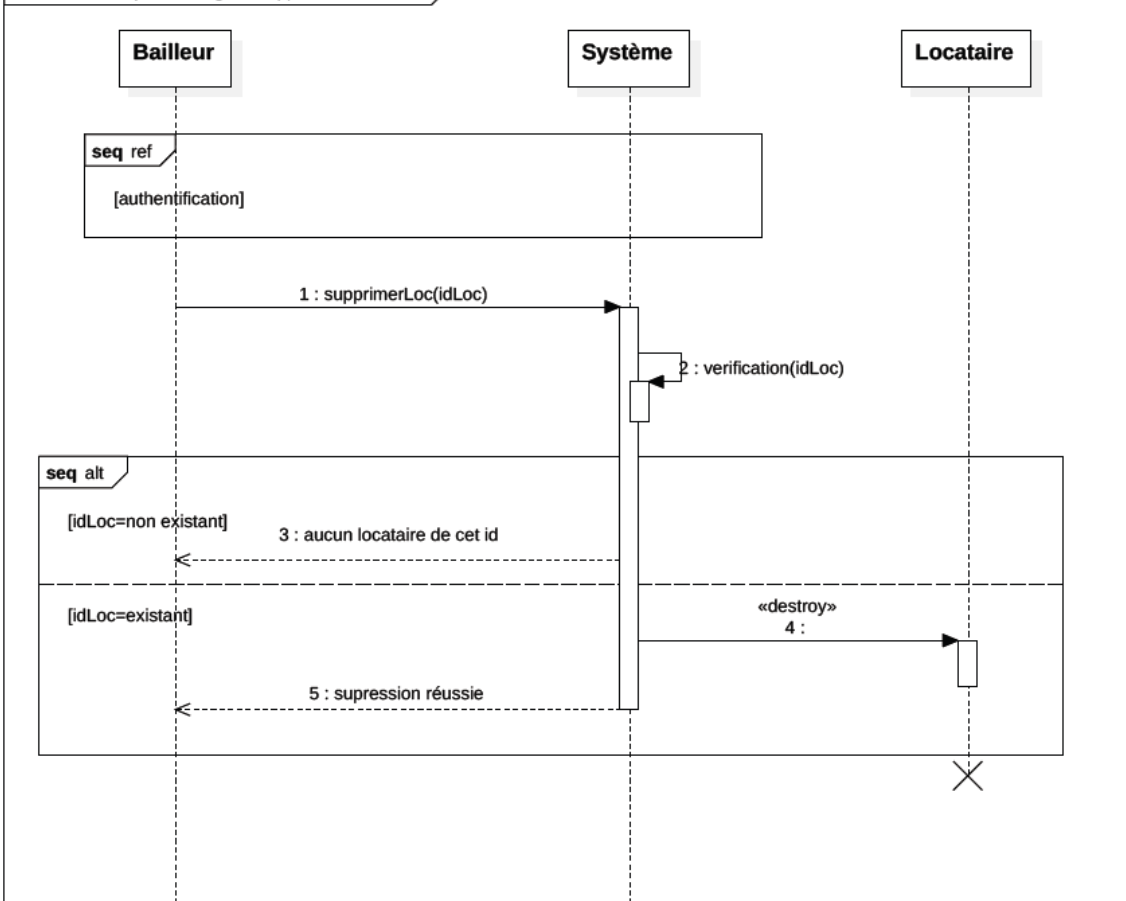
Collaboration5::Interaction1::SequenceDiagramauthentification



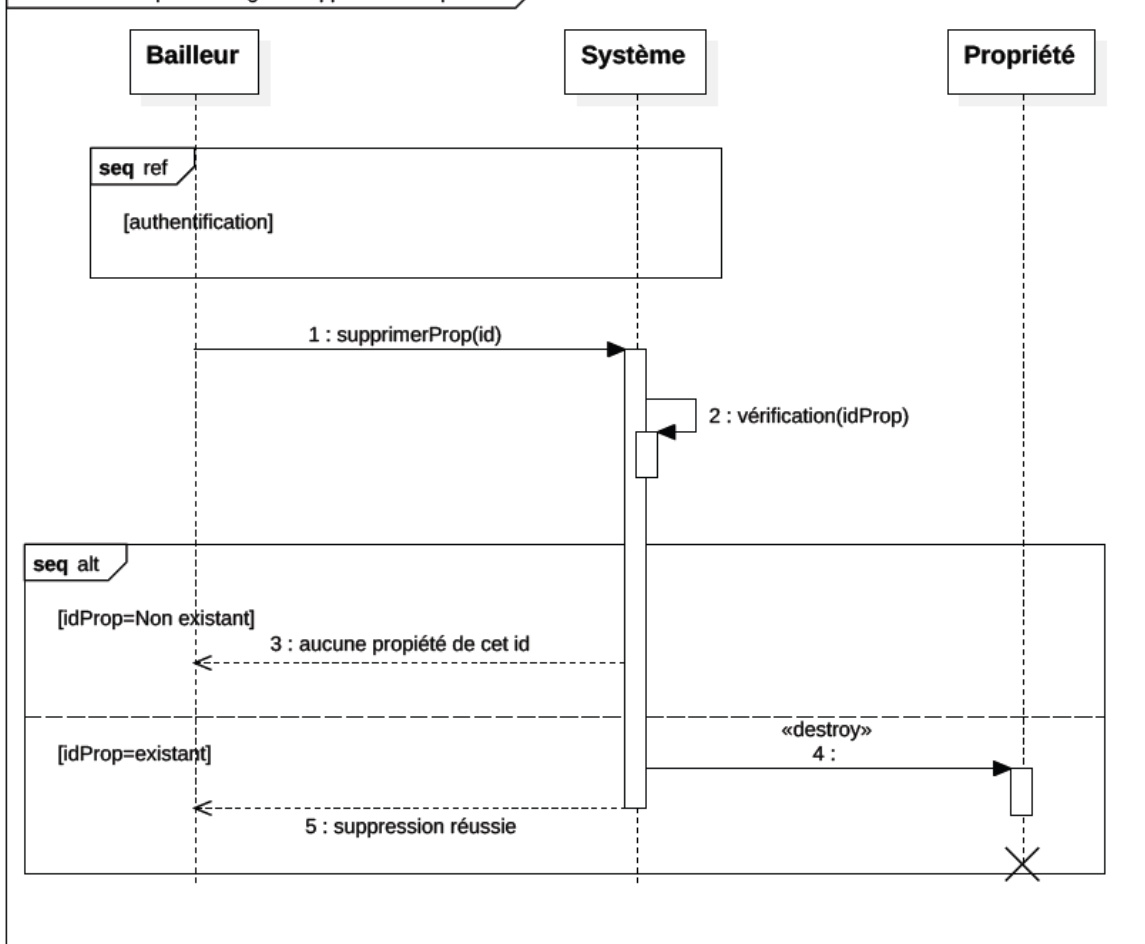
Collaboration1::Interaction1::SequenceDiagramAjoutLocataire

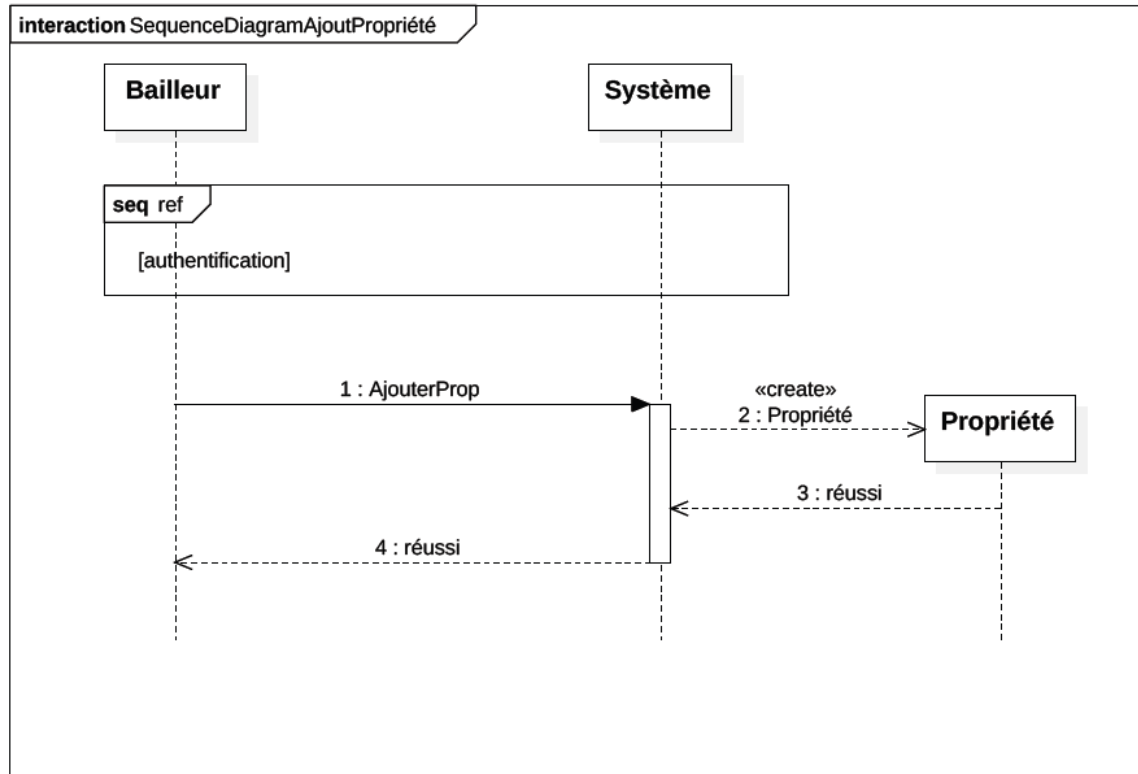


interaction SequenceDiagramSupprimerLocataire

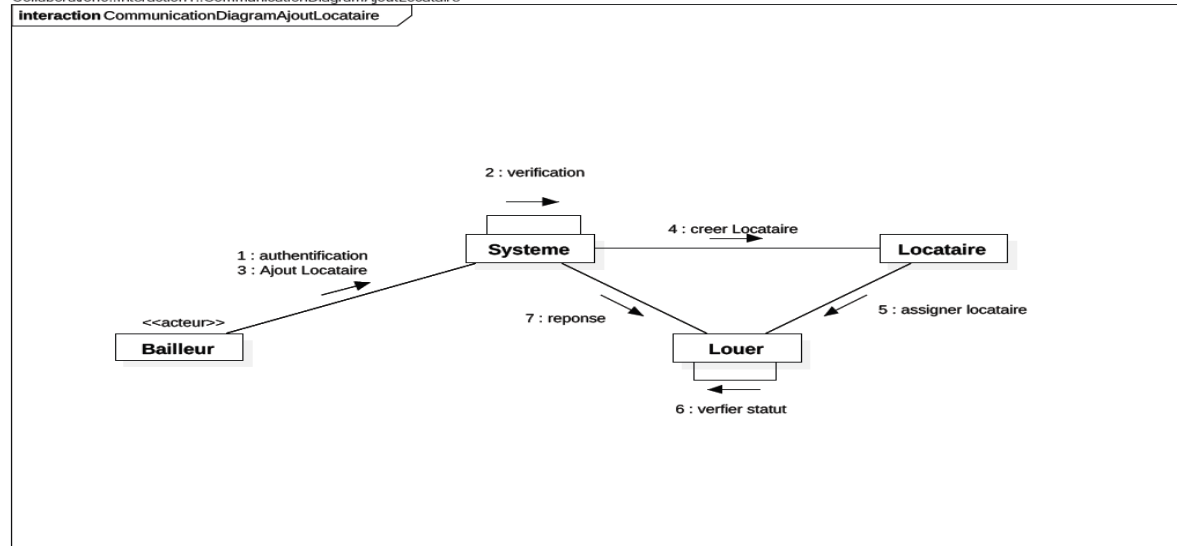
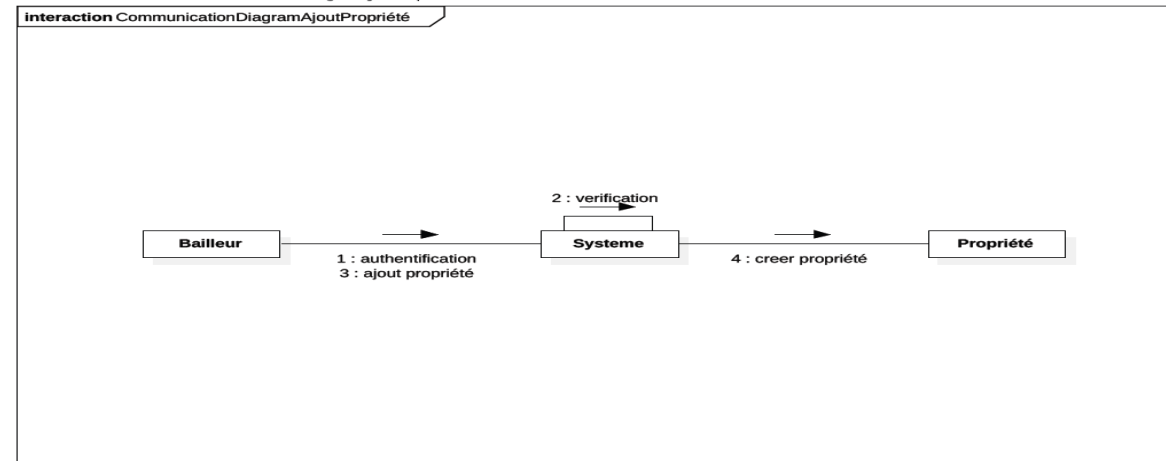


interaction SequenceDiagramSuppressionPropriété





6. Diagrammes de collaboration



Test de l'interface graphique

ID	Test	Technique de test
Test1	Démarrage : affichage de la fenêtre	Lancement du programme
Test2	Authentification	Entrer le nom et le mot de passe
Test 3	Choisir l'action à mener	Choisir dans le menu déroulant
Test4	Exécuter l'action choisie	Différent selon l'action choisit
Test5	Fermeture du programme	Fermer la fenêtre

IV. DEVELOPPEMENT

Lors du développement nous avons utilisé comme architecture logiciel le modèle MVC (Modèle-Vue-Contrôleur). Nous avons donc créé trois (03) packages :

1. Modèle

Celui-ci contient les données à afficher, ceci au travers des différentes classes qui y sont implémentés, entre autre LOCATAIRE, FACTURE, PROPRIETE, ... La classe principale ici est la classe Locataire, celle-ci permet de créer un objet locataire.

Dans le but de gérer la communication, il y a des importations que nous avons effectuées.

2. Vue

Ici est présentée l'interface graphique. L'interface graphique a été faite en utilisant Swing qui fait partie de la bibliothèque de Java Foundation Classe (JFC). Nous avons donc utilisé NetBeans qui nous a facilité le travail.

3. Contrôleur

Ce package contient les classes concernant la logique les actions effectuées par l'utilisateur. Modifie les données du modèle et de la vue. Elle communique donc avec la base de données, qui est un fichier .sql. Afin de mieux gérer cette connexion nous utilisons ici le build path mysql-connector-java-8.0.27.jar

V. BILAN DU PROJET

1. Apports individuels et collectifs

Il a été très difficile pour nous de réaliser ce projet, étant novices dans la plupart des outils particulièrement github. Nous avons dû nous mettre à l'apprentissage des outils tel que swing, asseoir la notion de MVC, et gérer la base de données la difficulté ici fût au niveau de la gestion des requêtes, donc la communication avec l'interface graphique.

2. Conclusion générale

Du point de vue de notre cahier de charge, ce projet est partiellement une réussite, la majeure partie de nos objectifs principaux ont été atteints : nous disposons de plusieurs méthodes, implémentant successivement des optimisations supplémentaires. Cependant, il reste encore beaucoup de choses qui pourraient être intéressant d'essayer.