|  |
| --- |
|  |
| Gestion Agence Immobilière |
| Document d'architecture et de conception |
|  |
| **HÉMERY Tugdual, VERGNAUD Romain, ROQUE Christopher, PARENT Vincent** |
| **Date (08-03-2018)** |

|  |
| --- |
|  |

# Suivi des changements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Numéro de version** | **Date**  (jj-mm-aaaa) | **Numéro de figure, table ou section** | **Type de changement** | **Description du changement** | **Originateur de la demande de changement** |
| **0.1** | 08.03.18 |  | Ajout | Ajout diagrammes et modèle MCD | themery |
| **0.2** | 11.03.18 |  |  | Ajout d’exigences | vparent |
| **0.3** | 15.03.18 |  | Modification  Ajout | Modification  MCD, MCC, MCT  Ajout commentaires | themery |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table des matières

[Suivi des changements 2](#_Toc508895955)

[Table des matières 3](#_Toc508895956)

[1.1 Description du projet 4](#_Toc508895957)

[1.2 Buts de ce plan 4](#_Toc508895958)

[1.3 Définitions, abréviations et acronymes utilisés dans ce document 4](#_Toc508895959)

[1.4 Hypothèses associées à ce plan 4](#_Toc508895960)

[1.5 Références 4](#_Toc508895961)

[2. Exigences 5](#_Toc508895962)

[2.1 Exigences fonctionnelles 5](#_Toc508895963)

[2.2 Exigences non-fonctionnelles 5](#_Toc508895964)

[3. Contraintes de conception 6](#_Toc508895965)

[4. Modèle conceptuel de communication (MCC) 7](#_Toc508895966)

[4.1 Modèle : 7](#_Toc508895967)

[4.2 Commentaire : 8](#_Toc508895968)

[5. Modèle conceptuel des traitements 9](#_Toc508895969)

[**5.1 Modèle :** 9](#_Toc508895970)

[**5.2 Commentaire :** 11](#_Toc508895971)

[6. Modèle conceptuel des données 12](#_Toc508895972)

[6.1 Modèle : 12](#_Toc508895973)

[6.2 Commentaire : 12](#_Toc508895974)

[7. Interfaces 13](#_Toc508895975)

[7.1 Interfaces utilisateur 13](#_Toc508895976)

[7.2 Interfaces matérielles 13](#_Toc508895977)

[7.3 Interfaces logicielles 13](#_Toc508895978)

[7.4 Interfaces de communications 13](#_Toc508895979)

[8. Matrice de traçabilité 14](#_Toc508895980)

**1. Introduction**

## 1.1 Description du projet

L’agence immobilière Dupont & Dupond veut une base de données adaptée à leur mode gestion. Celle-ci doit pouvoir gérer les biens qui sont proposés, les différents locataires et propriétaires de biens, enregistrer les différents loyers.

## 1.2 Buts de ce plan

Ce document explique les choix et étapes de conceptions de la base de donnée demandée par l’agence Dupont & Dupond.

## 1.3 Définitions, abréviations et acronymes utilisés dans ce document

|  |  |
| --- | --- |
| **Sigle** | **Description** |
| **HPT** | Hypothèses |

## 1.4 Hypothèses associées à ce plan

[Énumérez les hypothèses que vous avez utilisées lors de la rédaction de ce document. Les hypothèses constituent un ensemble de règles d’affaires que vous avez définies afin de compléter celles déjà soumises par vos enseignants.   
  
Identifiez et numérotez les hypothèses en utilisant le préfixe HPT.]

## 1.5 Références

[Cette section doit énumérer tous les documents qui sont cités en référence dans votre plan. Spécifiez pour chaque document son titre, son ou ses auteur(s), la date ainsi que toute autre information pertinente.   
  
Par exemple, il peut s’agir de normes qui ont été utilisées lors de la rédaction de ce document, de livres que vous avez consultés afin de mieux connaître certaines technologies relatives à votre projet ou de sites web contenant de l’information sur UML.  
  
Identifiez et numérotez les références en utilisant le préfixe REF.

Exemples :

Ref-1 : SEL (version 1.0), projet XYZ, Acme Inc, Département du développement logiciel, décembre 2011

Ref-2 : Document d’architecture (version 1.5), projet XYZ, Acme Inc, Département du développement logiciel, janvier 2012]

# 2. Exigences

## 2.1 Exigences fonctionnelles

[Cette section doit énumérer toutes les exigences fonctionnelles liées à votre logiciel. Soyez précis. Généralement, ces exigences constituent les fonctions du logiciel (ce qu’il doit faire).

*Une exigence fonctionnelle est une exigence définissant une fonction du système à développer. Ce que le système doit faire.*

Dans le cas d’un logiciel effectuant la gestion des dossiers étudiants d’un, une exigence fonctionnelle pourrait être l’affichage d’un dossier étudiant ou bien le calcul de sa côte Z.

*Le système doit enregistrer la commande du client.*

*Chaque commande doit avoir un identifiant unique (ORDER\_ID)*  
  
Identifiez et numérotez les exigences fonctionnelles en utilisant le préfixe EF.]

EF1 Le système doit être capable de générer une annonce pour un journal.

EF2 L’agence doit pouvoir ajouter un nouveau bien à la location.

## 2.2 Exigences non-fonctionnelles

[Cette section doit énumérer toutes les exigences non-fonctionnelles liées à votre logiciel. Soyez précis. Généralement, ces exigences constituent des propriétés vérifiables que votre logiciel doit posséder. Elles sont habituellement liées à l’usage, à l’efficacité, à la fiabilité, à la maintenance ou à la réutilisation.   
  
Dans le cas d’une application web effectuant la gestion de la paye des employés d’une firme, des exemples d’exigences non-fonctionnelles pourraient être que l’application puisse supporter 10 connexions simultanées ou que l’application soit compatible avec les navigateurs Internet Explorer et Firefox et Safari.

*Le système doit permettre à l’utilisateur d’accéder au solde de son compte en moins de 5 secondes.*

Identifiez et numérotez les exigences non-fonctionnelles en utilisant le préfixe ENF.]

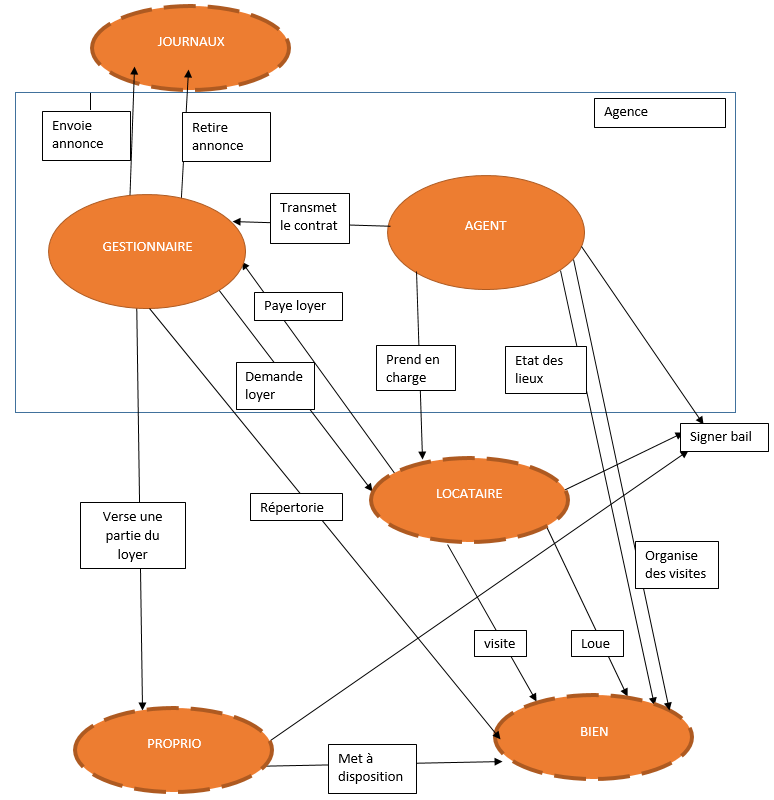
# 3. Contraintes de conception

[Décrivez les contraintes dont vous devez tenir compte dans la conception et la réalisation de votre logiciel. Les contraintes sont des restrictions affectant une partie de votre logiciel ou son entièreté.

Par exemple, une contrainte de conception pourrait être la nécessité d’utiliser le cadriciel (« framework ») Hibernate dans le cadre du développement d’une application Java ou de faire l’utilisation du serveur d’authentification corporatif du client.  
  
Identifiez et numérotez les contraintes en utilisant le préfixe CON.]

# 4. Modèle conceptuel de communication (MCC)

## 4.1 Modèle :



Flux

## 4.2 Commentaire :

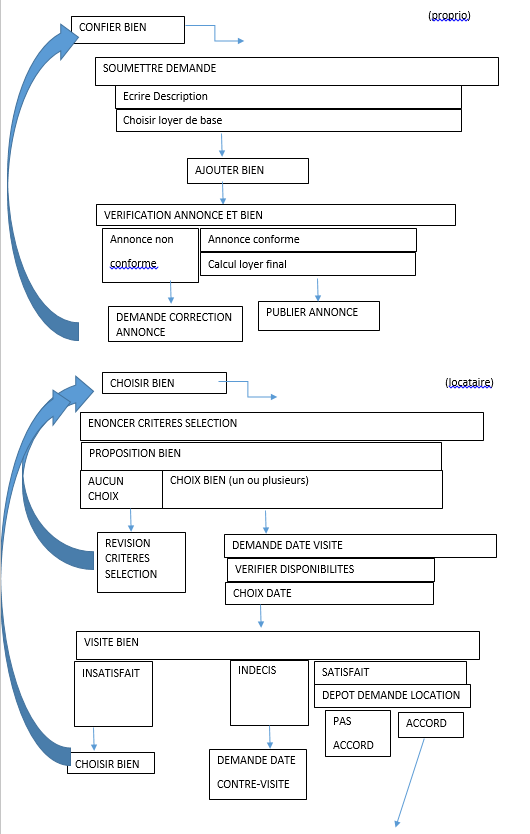
On a choisi de représenter le locataire, le propriétaire et le bien à l’extérieur de l’Agence. En effet ceux-ci n’appartiennent pas à la structure de l’agence.

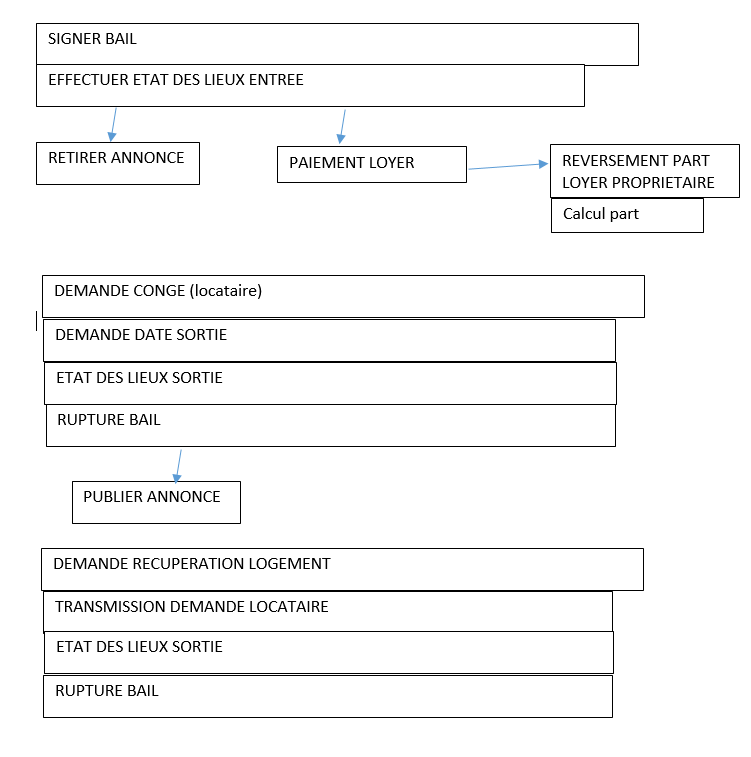
On a ajouté un acteur qui modélise toute les activités de gestion, notamment financière de l’agence. Cet acteur est responsable de réclamer et partager les loyers. Il émet également les annonces à destination des journaux.

Le rôle de l’agent a été revu, celui-ci sert maintenant d’intermédiaire entre le gestionnaire, le propriétaire et le locataire.

On a choisi de supposer que le rôle de l’agent et du gestionnaire vis-à-vis de l’agence est intrinsèque, en effet nous ne sommes pas supposés assumer connaître les méthodes de management de l’agence. Enfin ce nouveau modèle est plus souple, un agent est libre d’assumer la position de gestionnaire.

**5. Modèle conceptuel des traitements**

**5.1 Modèle :**

****

**5.2 Commentaire :**

Nous avons découpé le problème en 4 grandes fonctions : confier un bien, choisir un bien, quitter un bien et reprendre un bien. Chaque grande fonction est alors explorée de sa demande à son exécution, et est découpée en sous opérations et conditions.

Nous nous sommes d’abord intéressés à l’interaction PROPRIETAIRE/AGENCE car c’est de celle-ci que découle le processus de location du bien. Logiquement, nous avons ensuite considéré le processus de location par le locataire, de son choix de bien jusqu’à la manière de le quitter.

La réflexion sur le MCT à fait apparaître la nécessité de créer deux ‘loyers’, un loyer demandé par le propriétaire, et un loyer calculé et attribuée par l’agence.

Il fait également prendre conscience du problème de la gestion des emplois du temps des agents/employés pour la gestion des visites et les vérifications de disponibilités qui s’ajoutent.

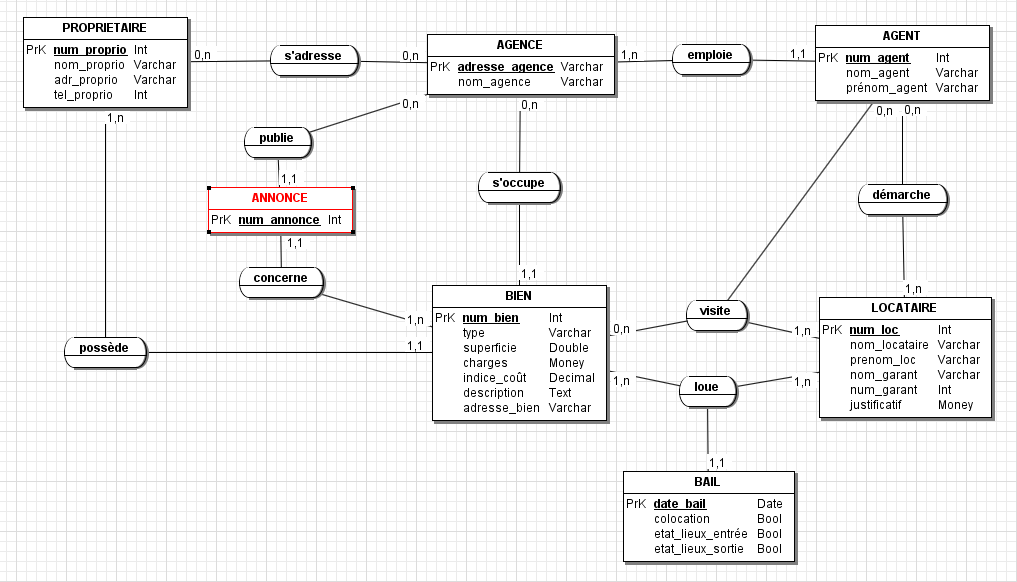
On remarque que certaines fonctions principales sont récursives, en fonction des conditions qui s’imposent lors de le traitement.

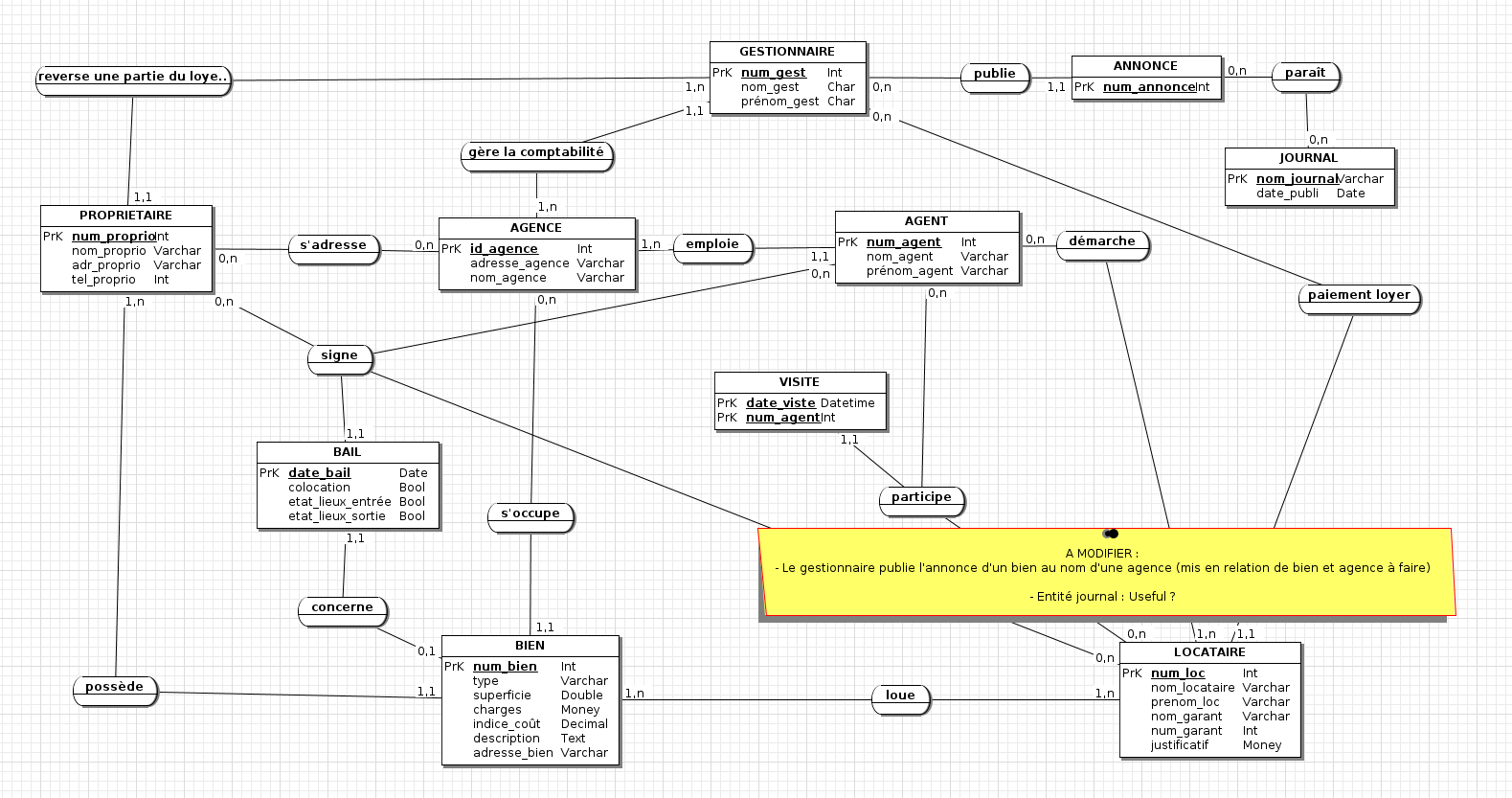
Ce MCT a été conçu en parallèle avec la version du MCC présente dans cette version du DSL, ce qui a permis de valider nos raisonnements une fois les deux modèles construits, et de donner la version corrigée du MCD présente dans cette version, qui est bien plus aboutie que les versions précédentes.

# 6. Modèle conceptuel des données

## 6.1 Modèle :

Version précédente :





Version actuelle :

## 6.2 Commentaire :

# 7. Interfaces

## 7.1 Interfaces utilisateur

[Spécifiez l’apparence de l’interface utilisateur de votre application. Il peut s’agir d’un descriptif textuel ou d’une image prototype. Si votre application contient plusieurs écrans importants, fournir plusieurs spécifications ou images.  
  
Identifiez et numérotez les interfaces utilisateur en utilisant le préfixe IU.]

## 7.2 Interfaces matérielles

[Spécifiez les interfaces que votre application peut avoir avec des composantes physiques. Ceci est surtout nécessaire dans le cadre de projet où l’application sera embarquée sur une composante physique, par exemple une machine à café ou un parcomètre.   
  
Identifiez et numérotez les interfaces utilisateur en utilisant le préfixe IM.]

## 7.3 Interfaces logicielles

[Spécifiez les interfaces de votre application peut avoir avec des composantes logicielles. Par exemple, il peut s’agir d’une base de données, d’un service web.   
  
Identifiez et numérotez les interfaces logicielles en utilisant le préfixe IL.]

## 7.4 Interfaces de communications

[Spécifiez les interfaces que votre application peut avoir avec des composantes de communication. Par exemple, il peut s’agir de l’Internet, d’un réseau privé (LAN) ou bien des ondes cellulaires.  
  
Identifiez et numérotez les interfaces logicielles en utilisant le préfixe IC.]

# 8. Matrice de traçabilité

[La matrice de traçabilité effectue les liens de référence entre les caractéristiques issues des besoins listés dans le document des spécifications logicielles et les cas d’utilisation de ce document.   
  
Le tableau suivant donne un exemple d'une matrice de traçabilité.]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Caractéristique / Besoin du client** | **Cas d’utilisation** | **Exigence fonctionnelle** | **Exigence non fonctionnelle** | **Contrainte** |
| Authentifier les utilisateurs | CU-1 | EF-1 | ENF-1 | CON-1 |
| Voir les dossiers étudiants | CU-2 | EF-2 | ENF-3 ENF-4 | CON-5 |
| Calculer cote Z | CU-3 | EN-5 | ENF-5 ENF-4  ENF-3 | CON-2 |