

Projet de fin d'année JAVA

Développement d'un logiciel de gestion de
vente immobilière

Michaël Silvestre



Ephéc 2012-2013

Table of Contents

1) Introduction.....	3
2) Technologie.....	3
Modèle.....	3
Controller.....	3
Vue.....	3
Utile.....	3
3) Modelisation	3
Groupe.....	3
Schéma relationnelle	3
Critique	5
4) Installation	5
Arborescence des fichiers et installation.....	5
5) Utilisation	6
Interface	6
Connexion d'un utilisateur	6
Création d'un bien.....	9
Création d'une offre.....	10
Gestion des demandes de visite, et des offres.....	11
6) Conclusion	13
7) Source.....	13

1) Introduction

Dans le cadre de ce projet, j'ai décidé de modifier quelque peu les consignes et de ne pas tout à fait les suivre à la lettre pour aller plus loin dans la recherche de développement d'une application fonctionnelle. Ceci dans le but de pouvoir m'aider à trouver par la suite un travail dans le secteur de la programmation objet en java et ainsi choisir une nouvelle orientation professionnelle. J'ai donc décidé unilatéralement de prendre quelque liberté dans le choix des modules, le style de codage et l'utilisation de l'application en tant que tel.

2) Technologie

Modèle

J'ai choisi d'utiliser un orm pour la gestion de ma base de donnée nommé : JeasyORM. Simple et léger il me permet de mapper directement mes objets avec mes tables en DB.

Controller

J'ai choisi également d'utiliser un moteur MVC pour gérer les déclenchements d'événements nommé : java-simple-mvc. J'ai rassemblé mes packets pour qu'ils correspondent à la pattern MVC. J'ai dû y adjoindre Slf4j pour la gestion du Log d'événements.

Cet MVC permet d'enregistrer des écouteurs globaux et de couvrir toute l'application via des modèles.

Vue

D'autre part, j'ai utilisé plusieurs modules graphiques.

J'ai commencé à designer mes pages à la main, puis j'ai opté pour jGoodies pour la conception de formulaires. Qui nécessite d'utiliser les bibliothèques : forms, common, bind

J'ai choisi un DatePicker : jcalendar

Utile

J'ai ajouté un module d'ouverture d'image avec preview

Et deux utilitaires. Un pour la copie de fichier et l'autre pour faire la génération de range de nombre

3) Modelisation

Groupe

Deux groupes de modèles existent. Les modèles DB et les modèle MVC.

Les modèles DB font le lien entre les entity et les tables en DB

Les modèles MVC font le lien entre le lancement d'un événement et la réception de ces événements pour l'ensemble de l'application à l'écoute.

Schéma relationnelle

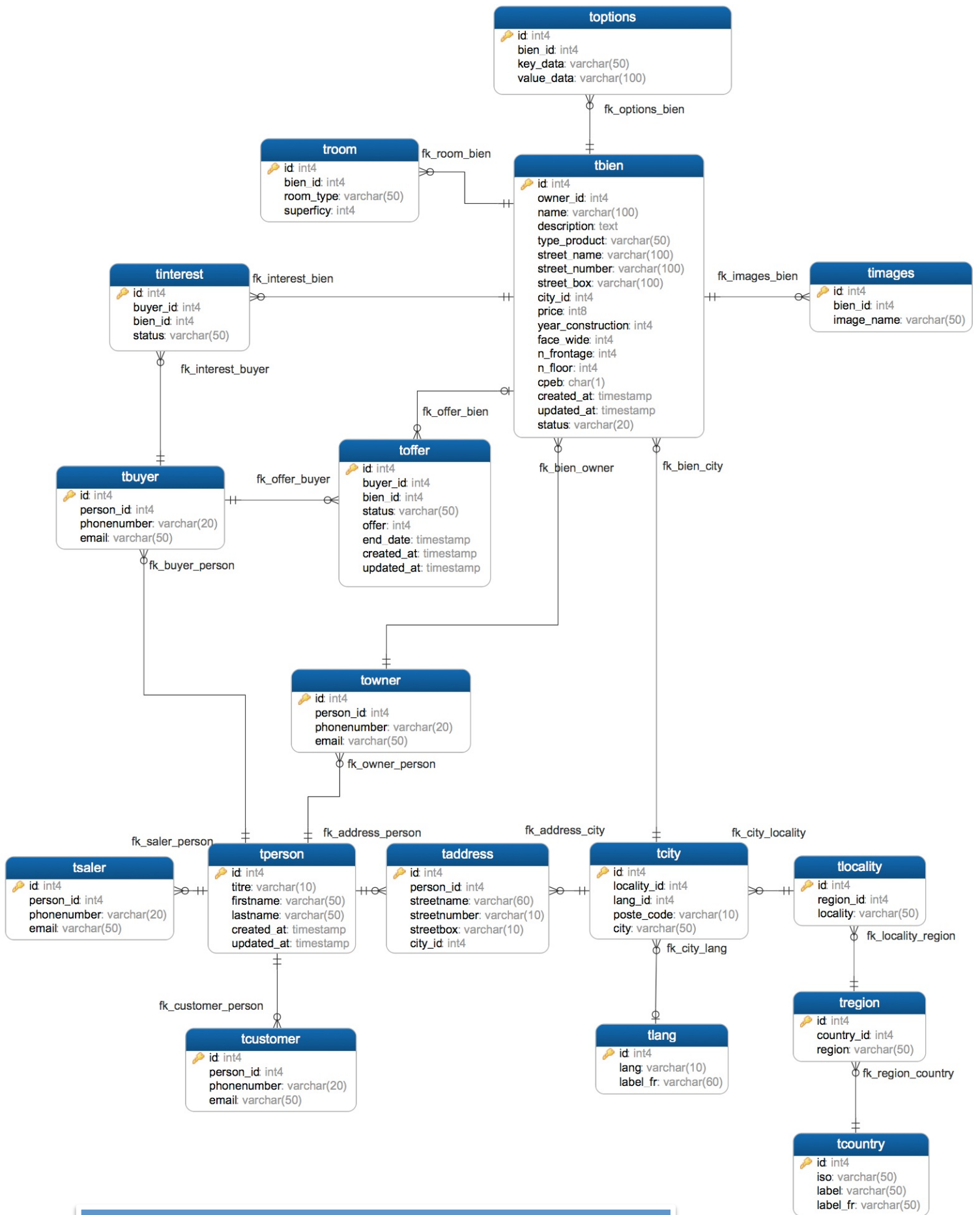


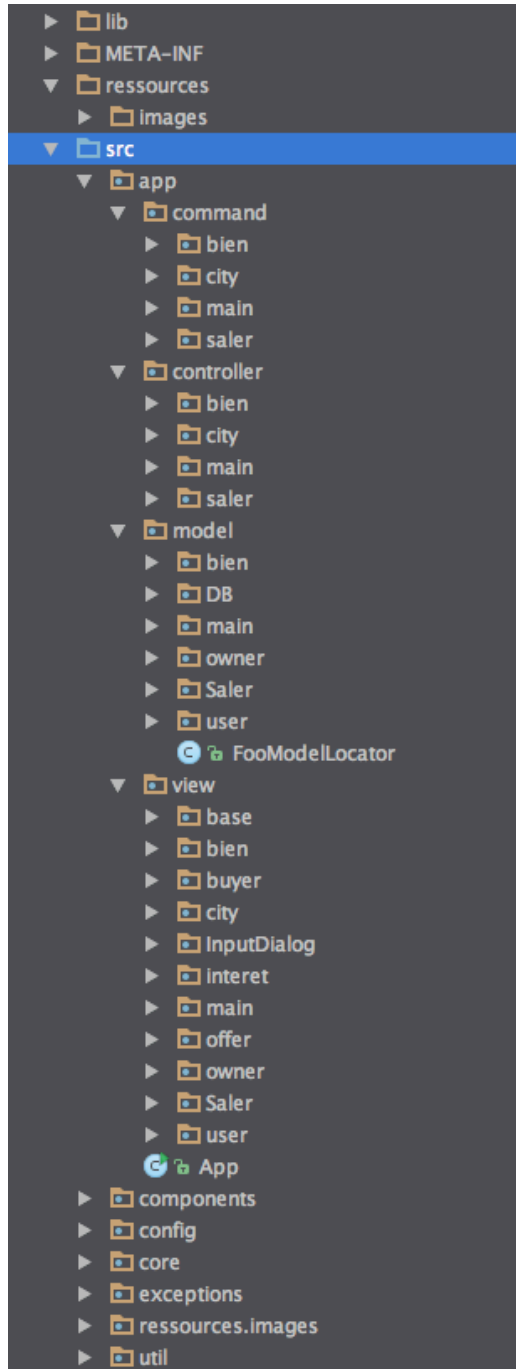
Schéma relationnel de l'application Immosales

Critique

Comme mis en évidence par le schéma relationnelle, le vendeur n'est pas relié à un bien. Un mécanisme de liaison devrait exister entre le bien et l'agent immobilier. A partir du moment où un bien est pris en charge par un agent immobilier, celui-ci devrait être lié avec le bien. Ce n'est pas le cas pour une raison de temps, mais un champ devrait être ajouté au bien avec un lien unaire à l'agent immobilier

4) Installation

Arborescence des fichiers et installation



Le projet est un projet IntelliJ12 (Idea)
Les formulaires sont développés par le framework idea intégré. On ne peut donc pas compiler le projet sans IntelliJ12 sous licence Community

Le dossier librairie contient tous les modèles nécessaire à l'utilisation du programme.

Le dossier ressources contient les images de l'application.

Le dossier src contient tous les packets nécessaires à l'utilisation du programme :

- app
- Components
 - o Composant importé
- Config
 - o Fichier de configuration
- Core
 - o Session
 - o EntityManager
- Exceptions
- Ressources
 - o images
- Util
 - o Utilitaires de l'app.

L'arborescence contient également un dossier DB avec les DB au format SQL originel au niveau structure et une base de donnée backup de démo
Il suffit d'exécuter un des fichiers SQL au choix dans une base de donnée Postgres et de modifier le fichier de configuration dans Config pour installer le programme.

5) Utilisation

Interface

Immosales est développé comme étant 3 applications distinctes. Une application par contexte. Ces trois applications seraient connectées à la même base de données. Une application chez le vendeur, une application chez l'acheteur et une application chez l'agent immobilier. Ces trois contextes coexistent dans l'application car un acheteur peut être un vendeur, un vendeur un agent immobilier, etc... mais ces trois utilisations sont bien des parties distinctes de l'application.

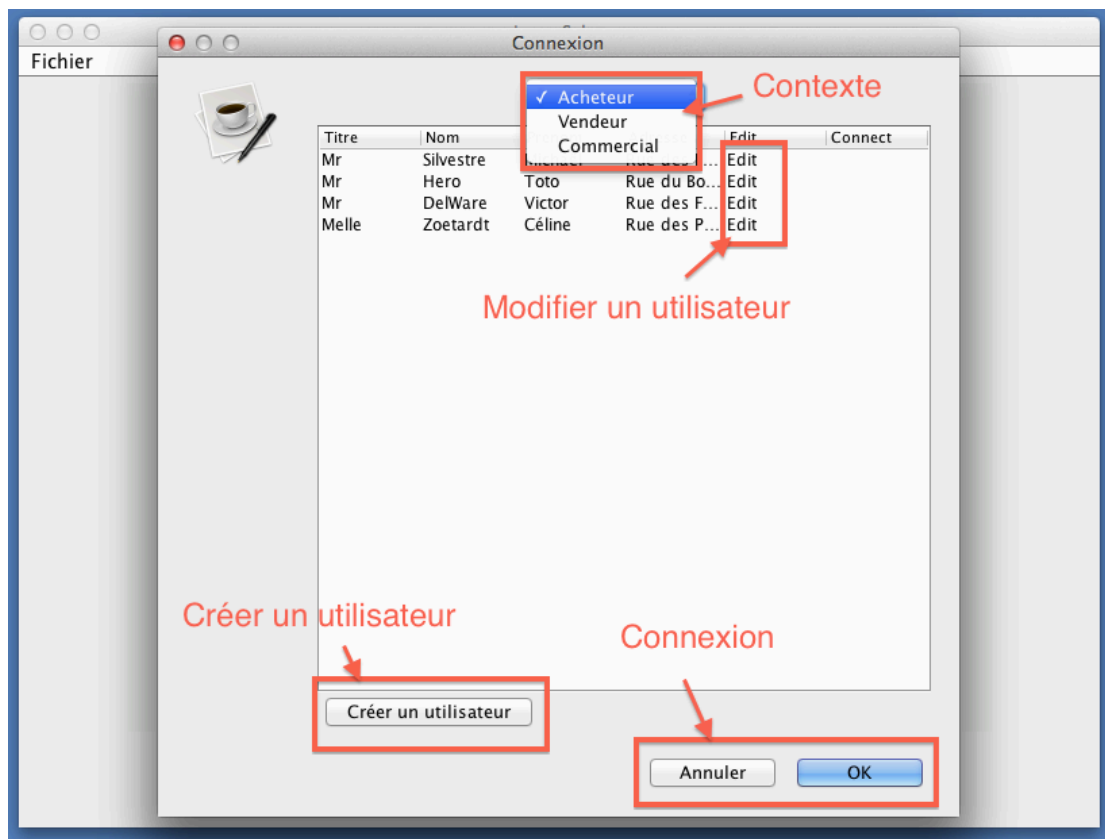
Ces trois contextes sont : Acheteur (Buyer), Vendeur (Owner) et Agent immobilier (Saler)



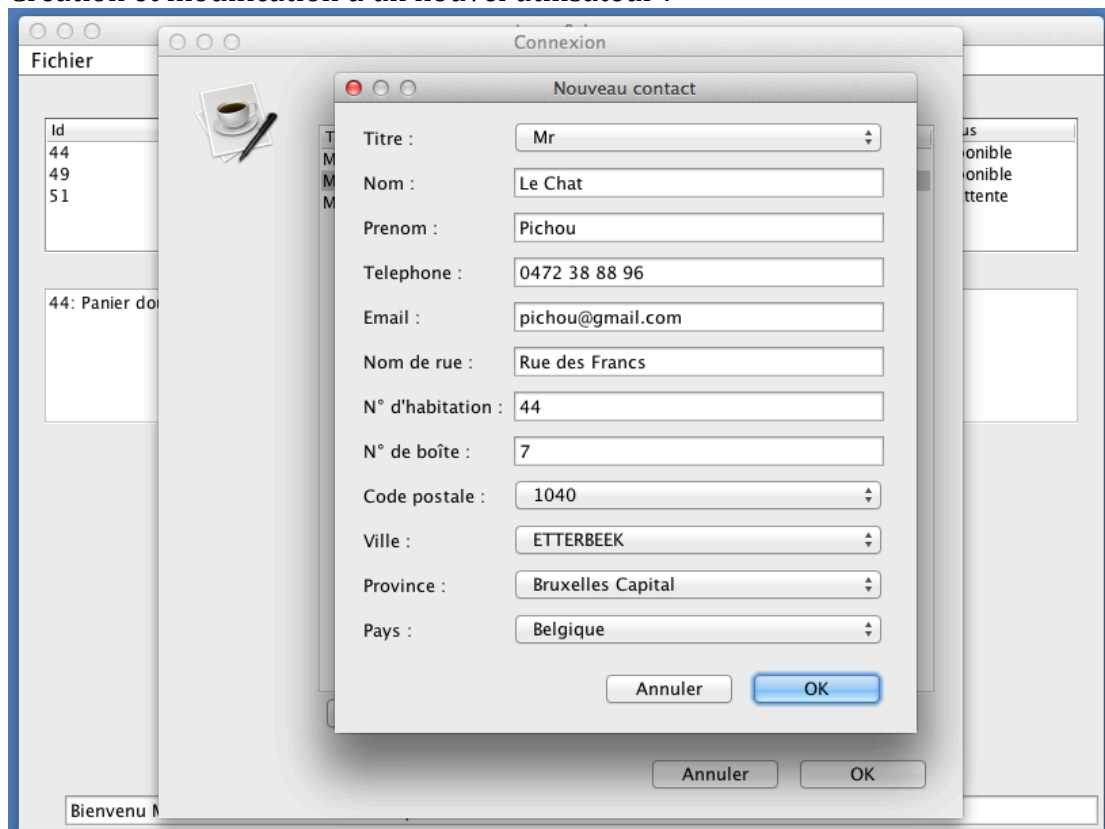
Connexion d'un utilisateur

Dès lors, à l'initialisation de l'application, l'utilisateur est invité à se placer dans un contexte avant de pouvoir aller plus loin.

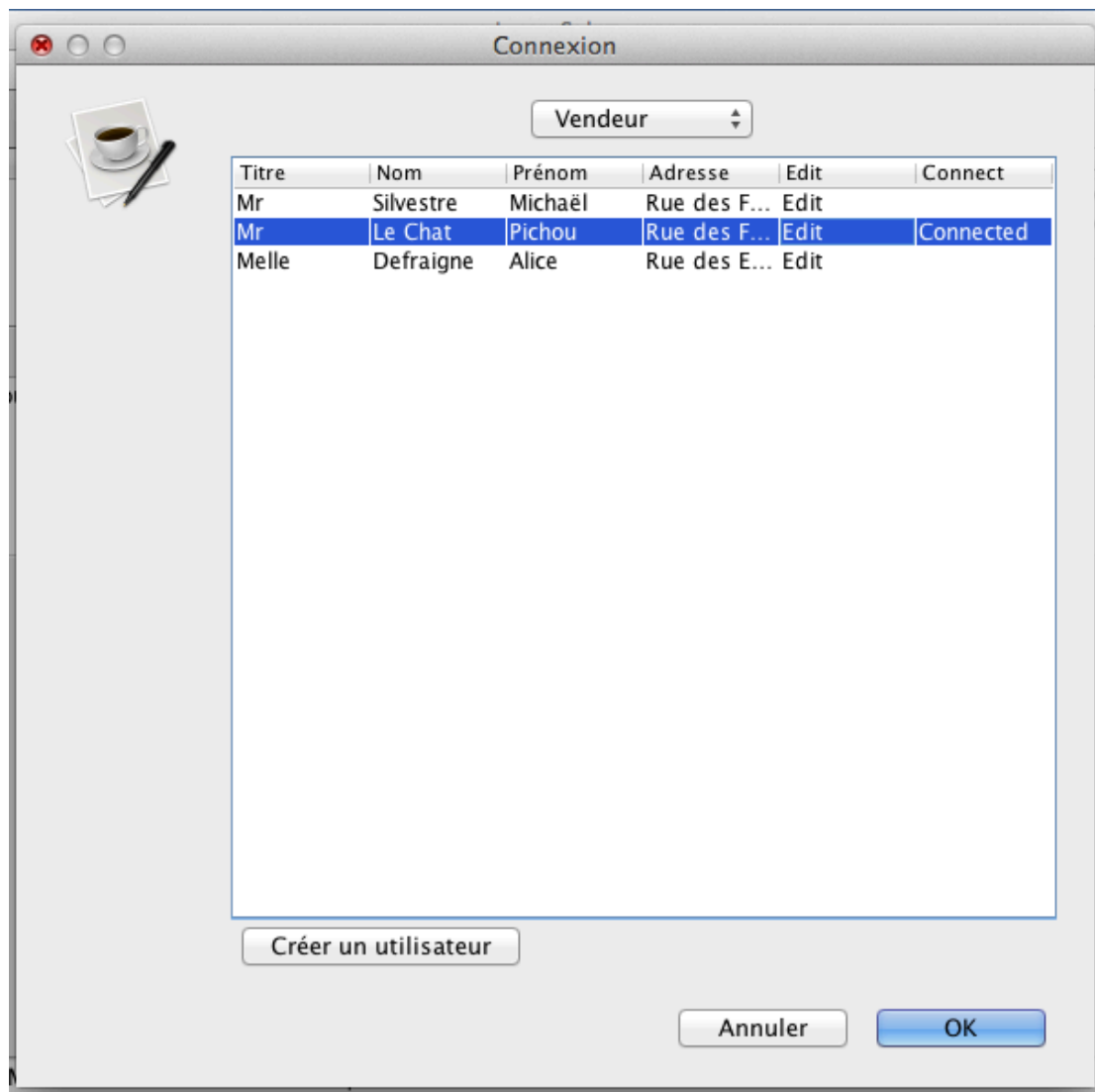
L'utilisateur doit cliquer sur Fichier puis connexion pour arriver à l'invite de connexion. Il faudra ensuite soit créer un utilisateur, soit sélectionner un utilisateur présent dans l'invite de connexion



Création et modification d'un nouvel utilisateur :



Etat de connexion d'un utilisateur lorsque l'on revient dans le panneau de connexion:



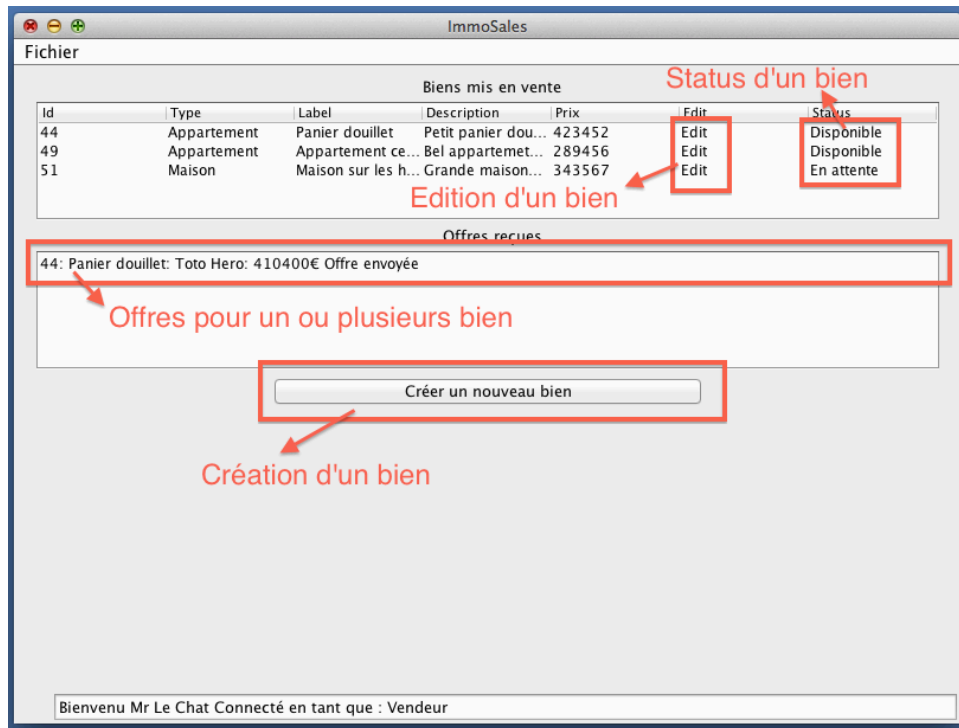
Si on est déjà connecté en tant qu'utilisateur et que l'on clique sur créer un utilisateur, on est invité à récupérer les données de l'utilisateur courant et de les appliquer sur le nouvel utilisateur pour ne pas perdre de temps.

Création d'un bien

Pour créer un bien il faut utiliser ou créer un utilisateur de type vendeur. En reprenant un utilisateur vendeur existant, on verra l'interface auquel il a accès.

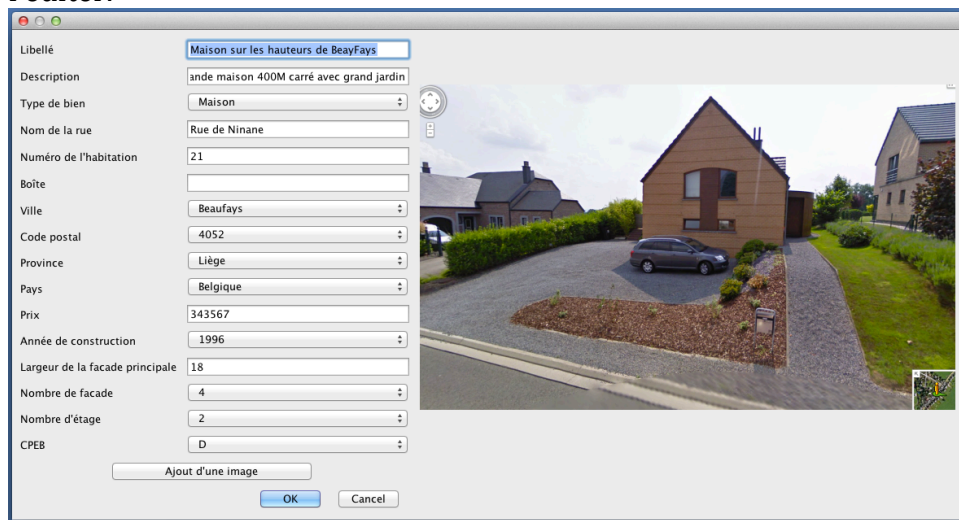
On peut voir le status du bien, créer ou éditer un bien, voir les offres faites pour tous les biens et si on sélectionne un bien on ne voit que l'offre pour ce bien.

En double cliquant sur un bien où une offre, le propriétaire peut voir celle-ci en détail. Il peut ainsi refuser une offre qui est faite sur le bien. Il pourra également signaler que l'acte d'un bien est signé



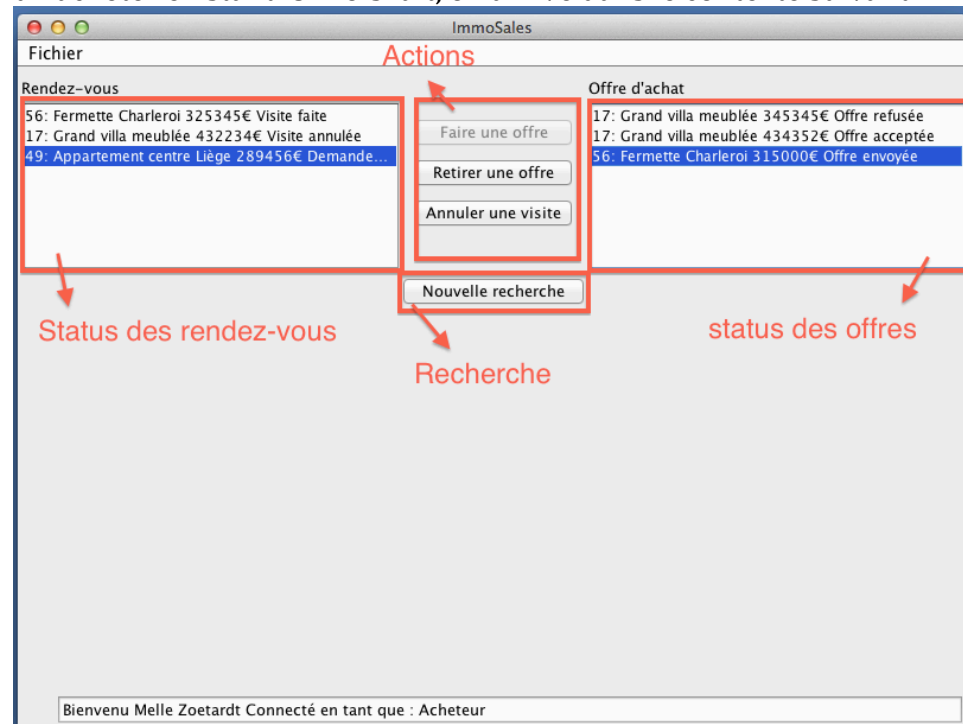
Création et édition d'un bien.

Le propriétaire peut éditer le bien tant qu'il n'est pas mis en ligne. Il peut également ajouter une photo du bien. Dès que le bien est en ligne il ne peut plus l'éditer.



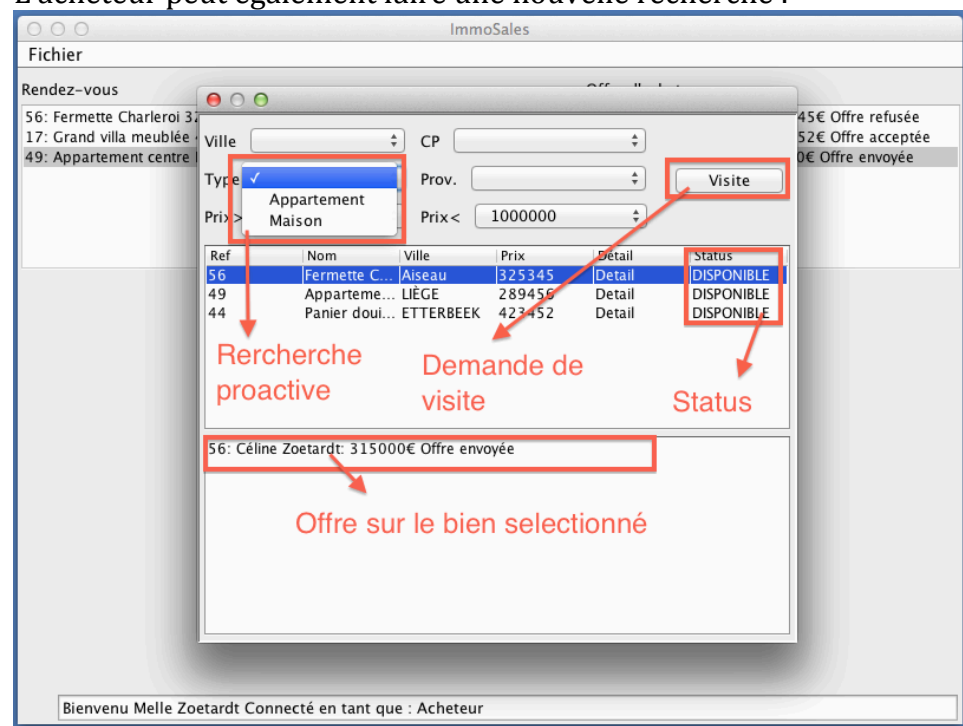
Création d'une offre

Il faut se placer dans le contexte Acheteur. Pour se faire il faut créer ou utiliser un acheteur existant. Un fois fait, on arrive dans le contexte suivant :



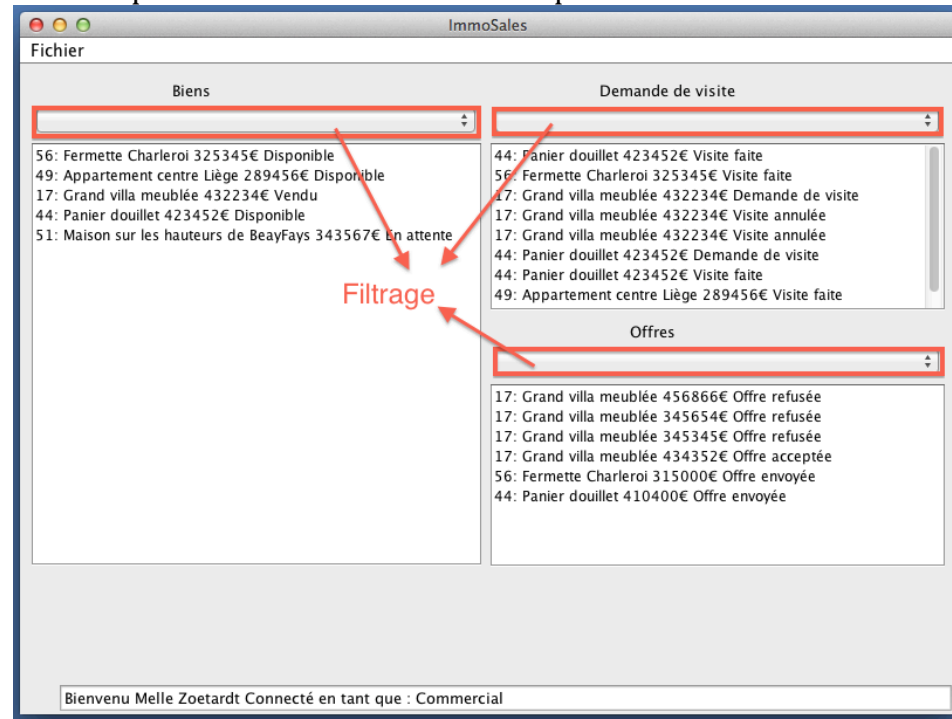
l'acheteur peut voir le statut des rendez-vous et des offres. Plusieurs actions sont possibles suivant leurs statuts. Annuler un rendez-vous, retirer une offre, faire une offre

L'acheteur peut également faire une nouvelle recherche :

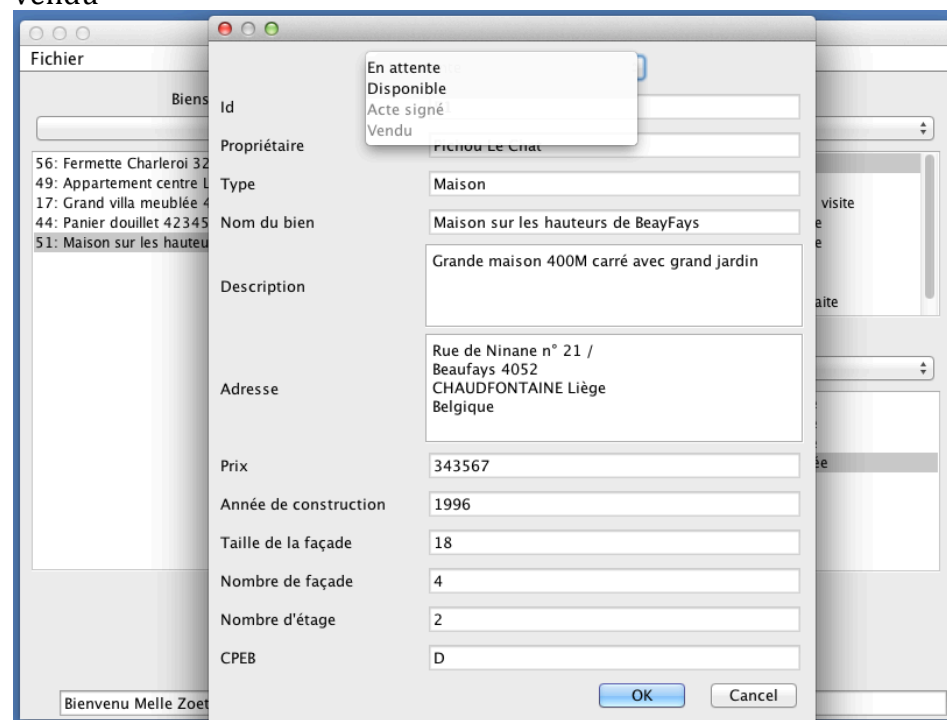


Gestion des demandes de visite, et des offres

La gestion des demandes et des offres est faite par l'agent immobilier. L'application doit être placée dans un tel contexte en créant et/ou utilisant un agent immobilier existant. Cet agent accède à toutes les transactions de l'application. Les biens mis en ventes et en attente de visite de l'agent immobilier, des demandes de visite d'un bien immobilier de la part d'un acheteur et d'accepter ou refuser des offres faites par des acheteurs sur un bien. :



L'agent fait passer un bien de l'état de bien en attente en bien disponible pour tous les acheteurs. Une fois l'acte signé il peut également signaler le bien comme vendu



L'agent peut organiser une visite du bien à l'acheteur

Fichier

Id

Acheteur potentiel: Victor DelWare

Propriétaire: Pichou Le Chat

Type: Appartement

Nom du bien: Panier douillet

Description: Petit panier douillet plein de poils car c'est rigolo

Adresse: Rue des Francs n° 44 / 7
ETTERBEEK 1040
ETTERBEEK Bruxelles Capital
Belgique

Prix: 423452

Année de construction: 2009

Taille de la façade: 20

Nombre de façade: 2

Nombre d'étage: 2

CPEB: A

OK Cancel

L'agent peut finalement ajouter une offer, la refuser ou annuler l'offre

ImmoSales

Status: Offre envoyée

Client: Céline Zoetardt

Propriétaire: Alice Defraigne

Offre: 315000 €

Bien: Nom: Fermette Charleroi
Description: Petite Fermette du côté de Charleroi
Adresse: Rue Grande n° 50 /
Aiseau 6250
AISEAU-PRESLES Hainaut
Belgique
Prix: 325345 €

OK Cancel

Bienvenu Melle Zoetardt Connecté en tant que : Commercial

6) Conclusion

Cette application se veut évolutive. J'ai pu approcher Java comme jamais avant cela. J'ai commencé ce projet au début du mois d'avril et je le termine en écrivant ces lignes fin juin. En écrivant les commentaires sur le code et en repassant celui-ci en revue je me rend compte du chemin parcouru. J'aurais tant de chose à refactorer... mais finalement c'est le propre de l'éducation. Se donner du temps pour apprendre un nouveau langage et pouvoir peut-être un jour en vivre. Je sais qu'il manque certaines choses basiques que je n'ai pas eu le temps de développer, mais je suis sûr de faire mieux la prochaine fois.

7) Source

<http://jeasyorm.sourceforge.net/>
<http://code.google.com/p/java-simple-mvc/>
<http://www.jgoodies.com/>
<http://www.slf4j.org/>
<http://www.jetbrains.com/idea/>
<http://www.toedter.com/en/jcalendar/>