

**DNET**

Scrabble

*Rapport*

Debecker Bernard

Foncier Romain

2012 - 2013

# Introduction

Dans le cadre du cours de *Développement InterNET* dispensé en dernière année de bachelier en informatique, il a été proposé aux étudiants de développer une application web de *location de biens de vacances* à l'aide du Framework Java *Spring*.

Le développement est à réaliser par groupe de 2 à 3 personnes. Le notre est composé des membres suivants : *Bernard Debecker* et *Romain Foncier*.

Ce rapport a pour but de présenter notre travail. Il comportera trois parties. La première décrira le travail d'analyse mis en place avant et pendant le développement. Les deux parties suivantes décriront respectivement les structures établies côté serveur, puis côté client, ainsi que les techniques et technologies mises en place de chaque côté.

# Analyse

Le développement d'un site web s'articule de manière générale autour de deux pôles principaux : serveur et client. Ces derniers se calquent sur le *cycle requête-réponse* à la base de l'architecture client/serveur d'une application web. Le client envoie une requête au serveur qui lui renvoie une réponse sous forme de données formattées en HTML, texte, JSON, XML, ...

Notre analyse a donc pris en considération cette interaction entre chaque partie, afin de très vite définir les informations qui allaient devoir être échangées, leur type et le contexte au sein duquel elles pourraient être consultées et modifiées.

L'application concerne un site de location de biens. Plusieurs questions ont donc été posées :

* Quelles informations définissent un bien ?
* Comment les utilisateurs peuvent-ils les consulter ?
* Quels vont être les différentes sections du site ?
* Quelles informations vont être liées à l'utilisateur et comment ?
* ...

Nous allons commencer la présentation de notre analyse par le modèle de données, puis nous nous intéresserons au design et aux cas d'utilisations.

### Modèle de données

Notre application propose un service de location de biens. Elle s'articule donc essentiellement autour de deux entités principales :

***Property***

Cette entité représente un bien. Afin de définir quels attributs doit posséder une propriété, il est important d'anticiper les informations qui seront affichées sur la plate-forme. Un utilisateur consultant la description d'une *property* devra y trouver les informations suivantes :

* **owner :** Propriétaire du bien (Foreign Key vers l'entité *user*)
* **title**
* **short\_desc** : Courte description affichée dans les résulats de recherche.
* **long\_desc** : Description détaillée du bien fournit par le propriétaire.
* **price**
* **type** : Type du bien : FLAT, LOFT, HOUSE, ...
* **rooms**
* **country**
* **city**
* **rent\_period\_start** : Date de début de location du bien.
* **rent\_period\_stop** : Date de fin de location du bien.

Un certain nombre de champs s'ajoute à cette liste venant y apporter de nombreuses informations complémentaires utiles à l'application et à certaines fonctionnalités :

* **added** : Date de création du bien.
* **modifed** : Date de dernière modification du bien.
* **coordinates** : Latitude et longitude du bien permettant d'utilisé *l'API Google Maps*.
* **note** : Moyenne des appréciations postées par les utilisateurs ayant loué le bien.

Il est important d'apporter quelques précisions concernant certain de ces attributs. Par simplification, nous considérons par défaut qu'une chambre (*room*) est disponible pour deux personnes. Un bien possédant deux chambres sera considéré par l'application comme pouvant proposer une location pour quatre personnes.

Afin d'apporter plus d'interactivité à notre site, nous avions dès le départ envisagé d'utiliser l'*API Google Maps*. Il nous est de cette manière possible de proposer une localisation d'un bien lors de la consultation de celui-ci, mais également lors de la phase d'ajout. Le stockage des *coordonnées géographiques* (*lat/lng*) permet d'en faciliter l'utilisation et offre également des possibilités de recherches précises ou sur des zones de taille donnée. Ces éléments seront détaillés dans la section consacrée au système de recherche.

D'autre part un système de notation a été mis en place. Basé sur les critères d'appréciations suivants :

* **cleanliness** : Etat de propreté général du bien
* **confort** : Confort de la literie, des installations, ...
* **quality/price** : Rapport qualité-prix

Le processus de notation a lieu une fois le séjour passé. L'utilisateur à l'origine de la réservation sera invité à fournir ses données. Une moyenne de l'ensemble de ces critères pour un bien permettra d'en définir la *note globale*.

***User***

Cette entité est importante et indispensable au fonctionnement général de l'application. Elle est composée des attributs suivants :

* **username** : Unique sur la plate-forme.
* **first\_name**
* **last\_name**
* **email**
* **password**
* **created** : Date de création du compte.
* **is\_staff** : Booléen spécifiant si l'utilisateur a accès à l'interface d'administration.

*User* est une entité clé. Une *property* ne peut être créée sans utilisateur. Si aucune propriété n'est définie, l'application ne peut fonctionner correctement.

Lien entre user et ses autres tables ...