**PROJET**

Dans le but de mener à bien notre projet, nous faisons une analyse détaillée de celui-ci, ceci afin de mieux comprendre les besoins de notre client (MOA), et de pouvoir trouver une solution qui lui conviendra. Ce document nous permettra donc de faire une conception fonctionnelle, une conception technique.

* PRESENTATION DU PROJET
* **Titre du projet**

LOGICIEL DE GESTION LOCATIVE S’OCCUPANT DE LA FACTURATION.

* **Beneficiaries**
* Les Bailleurs sont les principaux bénéficiaires de ce projet, en plus des locataires.
* Ce projet permettra donc au bailleur de connaitre le statut de chacun de ses locataires (qui paye quand ? combien ? qui a des aérés ? de combien ?...) d’autre part il permettra de rappeler aux locataires la date à laquelle ils doivent payer, et les alerter en cas d’aéré.
* **Contexte et enjeux du projet**
* L’un des membres de ce groupe étant fils de bailleur, a observé et a constaté les problèmes auxquels font face les bailleurs, surtout lorsque ceux-ci ont plusieurs locataires.
* Il constate alors qu’il est compliqué pour le Bailleur de se souvenir lequel des locataires a des aérés et de combien de mois, ou de se rappeler quand chacun de ceux-ci doit payer, bien même lorsqu’il possède des registres pour le faire, de plus certains locataires doivent toujours rappelés et prouvé au bailleur qu’ils ont payé, qu’ils n’ont pas de retard, etc...
* **Problème**
* Les bailleurs ayant beaucoup de propriétés (maison, boutiques,) sont très souvent confrontés à un problème de gestion de leurs locataires. Il est alors pénible de savoir **qui paye ?** Et **quand ?** Ou alors **qui a des aérés de loyer ?** Et **de combien ?**
* Du côté du locataire, il lui est parfois pénible de se rappeler **quand il est entré en location ?** A **quel moment il doit payer ?** Ou alors s’il a un **aéré lointain**…

* CAHIER DE CHARGE FONCTIONNELLE
* **Besoins fonctionnels**
* Permettre au bailleur d’exposer ses biens pour d’éventuels clients.
* Permettre au bailleur de facturer ses locataires.
* Rappel au Bailleur la date à laquelle chaque locataire doit payer, rappelle également le locataire en avance.
* Les différentes propriétés du bailleur sont enregistrées dans la base de données.
* Pour chaque locataire, le bailleur possède une trace de tous ses payements ainsi que toutes ses informations.
* Permettre un archivage des factures tant pour les locataires que pour le bailleur.
* 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.
* **Besoins Non fonctionnels**
* Sécurité : une fois que le bailleur à facturer le locataire, il ne doit plus faire marche arrière pour changer les données.
* Fiable : le système doit être capable de rappeler à temps les utilisateurs de la date de paye au moment favorable.
* Facile à utiliser : le logiciel doit être compréhensible et simple à manier.

* CONCEPTION TECHNIQUE
* **Objectifs et résultats attendus**

Le logiciel que nous proposerons aidera donc le bailleur à :

* Gérer ses propriétés (listes des propriétés libres, occupés, ajouter une propriété).
* Gérer ses locataires (ajouter un locataire, modifié, supprimé).
* Gérer les factures (archiver tous les payements des différents locataires, rappeler les éventuels aérés).
* Etc…

Le logiciel que nous proposerons aidera donc le locataire à :

* Communiquer avec le Bailleur.
* Lui rappeler la date à laquelle il doit payer.
* Etc…
* **Choix Techniques et justifications**
* Langage de programmation orientée objet **JAVA.**
* IDE **INTELLIJ.**
* Framework **SPRING BOOT, BOOTSTRAP, …**
* Utilisation de html, CSS.
* Nous avons choisi Java principalement parce que nous estimons qu’il sera plus aisé pour nous d’appliquer les concepts de la conception orienté objet en utilisant ce langage, de plus une de ses plus grandes forces est son excellente portabilité : une fois notre programme créé, il fonctionnera automatiquement sous Windows, Mac, Linux, etc… Bootstrap nous permettra d’avoir un site web responsive, donc utilisable sur tablette, PC, téléphone portable, etc...
* **Architecture**

Pour ce logiciel nous avons optés pour une application monolithique basée sur le patron de conception MVC. Ainsi les requêtes venant de la vue seront transmises à un composant appelé "dispatcher servlet" qui à son tour choisira le contrôleur adéquat. Ce contrôleur grâce à l’interface des web services que proposera la couche service fera appelle à la couche métier concerné afin que l'information de la vue soit traitée.

Après avoir traité l'information, la coucher métier retransmettra le résultat de son traitement à un moteur de Template (ici c'est Thymeleaf que nous utiliserons) qui génèrera la nouvelle vue coté serveur et la remettra à dispatcher servlet qui l'affichera au client.

* **Description des traitements**

Dans cette application nous disposons de plusieurs traitements classés selon différents modules :

-Le module de gestion de locataire consiste à ajouter, modifier, supprimer, attribuer une propriété a un locataire

-Le module de gestion de propriétés consiste à ajouter, modifier, supprimer, publier une propriété libre.

-Le module de gestion de factures consiste à attribuer une facture a un locataire suivant son payement au cours du mois ou chaque fin de mois, d’envoyer une facture au format PDF suivant les payements du locataire concerné.

-Le module de Messagerie consiste à l’envoie et réception de messages entre Bailleur et locataire.

* **Description des Interfaces**

Plateforme est constituée de nombreuse interfaces, mise sur pieds pour faciliter le plus possible son utilisation ; il est constitué de 03 grandes parties à savoir :

* **L’interface du bailleur,** cette partie est beaucoup plus réservé uniquement au bailleur, car c’est à partir de cette interface que lui seul pourra effectuer des actions dites importante dans notre plateforme notamment : **ajouter un locataire, supprimer un locataire, modifier un locataire, ajouter une propriété, supprimer une propriété, modifier une propriété, facturer un locataire.**
* **L’interface du locataire,** cette partie de notre plateforme concerne beaucoup plus le locataire, qui lui pourra à partir de cette page consulter par exemple : **ces factures (récentes ou anciennes), discuter avec le bailleur (**pour d’éventuelle location ou proposition**), consulter d’autres propriété.**
* **L’interface propriété,** cette partie de notre plateforme est beaucoup axé sur la présentation et la description de différents biens ou propriétés mise en location par le bailleur (**appartement, studio, chambre, maison, boutique, immeuble, terrain**), a nos différents visiteurs de notre plateforme et qui par la suite s’ils sont intéressés pourront entrer en contact avec le bailleur.
* **Liste de tâches**
* Maquettage des interfaces graphiques.
* Page d’accueil.
* Elaboration de la base de données.
* Formulaire d’identification du bailleur.
* Elaboration de l’interface du Bailleur.
* Fonctionnalité ajouter/supprimer locataire.
* Page messagerie pour chaque locataire.
* Module gérant la facturation.
* Test généraux.
* Déploiement.