

Projet Web Serveur - Rapport sur la réalisation d'un système de gestion d'un parc immobilier

Killian GALFRE - - VEYRET et Jacques KOZIK

Du 28 mars au 12 mai 2024

Table des matières

1	Introduction : Description général du projet	1
2	Réponses au TP1	1
3	Design URI de notre application	2

Résumé

Ce projet réalisé au cours des dernières semaines a consisté en la réalisation d'un site permettant de gérer un parc immobilier. Ce rapport contient un bref résumé de l'architecture de notre site.

1 Introduction : Description général du projet

Ce projet à été réalisé principalement en langage Java. Nous avons également utilisé du ftl pour réaliser nos templates et du H2 pour la gestion de notre base de données. Par ailleurs, il fait également appel à d'autres logiciels tel que Gitlab et son espace de travail partagé, ou encore le logiciel \LaTeX qui nous permet de rédiger ce rapport. Ce projet comporte plusieurs documents :

1. Le **code source** du projet réparti sur plusieurs fichiers .class
2. Les **templates** de nos pages .ftl
3. Un **rapport** sur le projet *GALFREVEYRET_KOZIK.pdf* fait à partir de \LaTeX

2 Réponses au TP1

Question 1 : Quelles sont les écrans (interfaces) qui vous semblent intéressantes à afficher aux utilisateurs ?

Voici les écrans à afficher :

- Liste des appartements d'un propriétaire / locataire
- Liste des immeubles / syndicats / personnes
- Liste des appartements par immeuble et des occupants par appartement
- Création d'un appart / immeuble / syndicat / personne
- Fiche "Identité" (pour une personne)
- Page de connexion / inscription
- Page de contact

Question 2 : Quelles vont être les principales classes de votre application ?

Voici les principales classes de notre application :

- Immeuble
- Syndicat
- Appartement
- Personne
- Occupation

Question 3 : Quelles vont être les tables de votre base de données ?

Voici les tables de notre base de donnée :

- Immeuble : nom, adresse, syndicat (FK vers syndicat¹)
- Syndicat : nomSyndic, adresse, nomReferent (FK vers Personne), numeroTel (FK vers Personne), adresseMail
- Appartement : etage, numero, superficie, estLoue, adresse (FK vers immeuble)
- Personne : nom, prenom, numTel
- Occupation (table de jointure entre Appartement et Personne) : numero (FK vers Personne), statut (Locataire / Propriétaire), numAppartement (FK vers Appartement), adresseAppartement (FK vers Appartement)

Question 4 : Quelles solutions technologiques envisagez vous ?

Voici les solutions technologiques utilisées pour ce projet :

- Java
- Spark Framework pour la gestion des requêtes HTTP et des routes
- Base de données (H2)
- FreeMarker (FTL) pour les templates
- HTML/CSS pour le front-end

3 Design URI de notre application

Connexion

- **GET** /connexion va afficher la page de connexion

1. FK = Foreign Key

- **POST** /**connexion** permet de se connecter

Inscription

- **GET** /**inscription** va afficher la page d'inscription
- **POST** /**inscription** permet se s'inscrire

Immeuble

- **GET** /**immeubles** va afficher la page listant tout les immeubles
- **GET** /**createImmeuble** va afficher le formulaire de création d'un nouvel immeuble
- **POST** /**saveImmeuble** va créer un nouvel immeuble
- **POST** /**deleteImmeuble** va supprimer un immeuble existant (et tout les appartements qui se trouvent dedans)

Appartement

- **GET** /**appartements** va afficher la page listant tout les appartements d'un immeuble
- **GET** /**createAppartement** va afficher le formulaire de création d'un nouvel appartement
- **POST** /**saveAppartement** va créer un nouvel appartement
- **POST** /**deleteAppartement** va supprimer un appartement existant

Syndicat

- **GET** /**syndicats** va afficher la page listant tout les syndicats
- **GET** /**createSyndicat** va afficher le formulaire de création d'un nouveau syndicat
- **POST** /**saveSyndicat** va créer un nouveau syndicat
- **POST** /**deleteSyndicat** va supprimer un syndicat existant

Personne

- **GET** /**personnes** va afficher la page listant toutes les personnes
- **GET** /**createPersonne** va afficher le formulaire de création d'une nouvelle personne
- **POST** /**savePersonne** va créer une nouvelle personne
- **POST** /**deletePersonne** va supprimer une personne existante

Occupation (lien entre une personne et un appartement)

- **GET** /**occupations** va afficher la page listant tout les locataires et propriétaires d'un appartement
- **GET** /**createProprietaire** (resp. **GET** /**createLocataire**) va afficher le formulaire de création d'un nouveau propriétaire (resp. locataire)
- **POST** /**saveProprietaire** (resp. **POST** /**saveLocataire**) va ajouter un nouveau propriétaire (resp. locataire) à l'appartement
- **POST** /**deleteProprietaire** (resp. **POST** /**deleteLocataire**) va retirer un propriétaire (resp. locataire) de l'appartement

Statistiques

- **GET** /**stats** va afficher la page permettant de consulter les statistiques d'un parc immobilier
- **GET** /**statsImmeuble** va afficher la page permettant de consulter les statistiques d'un immeuble