

# Visit My Cities

25/03/2020

---

Florent Bourgeois - Lionel Meyer

Licence Professionnelle Développeur d'Application Orientée Objets

123 Your Street

Your City, ST 12345

## Pourquoi ?

Dans le contexte actuel, il est délicat de proposer un contenu formatif traditionnel. En effet, l'absence du langage corporel nuit à la transmission de connaissances lorsque les apprenants et le formateur ne sont pas dans le même lieu.

C'est pourquoi nous vous proposons de modifier l'approche d'apprentissage proposé pour la période qui vient.

A partir d'aujourd'hui, nous passons en apprentissage en mode hybride par projet. Ces termes veulent simplement dire que vous allez avoir un projet à réaliser. Pour réaliser ces objectifs ils vous faudra acquérir les différentes compétences de la formation LPDAOO. Et pour les acquérir, nous vous pointeront des ressources en ligne et seront à votre disposition lorsque cela est nécessaire pour éclaircir les points qui posent des difficultés.

## Vue d'ensemble

Le projet Visit My Cities référence les différents bâtiments remarquables de chaque ville. Elle permet à un visiteur de les voir en fonction de ses préférences (année de construction, style, ...) pour lui permettre de prévoir les visites à faire lors de son passage dans la ville. Un utilisateur reconnu est également capable de renseigner de nouveaux bâtiments.

## Goals

1. Référencer l'ensemble des bâtiments notable
2. Pouvoir parcourir l'ensemble des bâtiments d'une ville
3. Permettre l'ajout et l'édition de bâtiments
4. Filtrer les bâtiments en fonction de critères spécifiques (type de bâtiment, années de construction, ?couleurs?, architecte, ...)
5. Prendre en charge différent types de bâtiments
6. Permettre de détailler un bâtiment en ajoutant des détails d'architectures spécifiques (toiture, charpente, fenêtres, ...)
7. Ajouter des bâtiments à la liste "à visiter"
8. Afficher un parcours avec les bâtiments à visiter
9. Permettre à un utilisateur de s'identifier

## Specifications

L'objectif du projet est de développer l'ensemble de l'application Visit My Cities. Cette application devra être réalisée sous la forme d'un (ou des) service(s) back et d'un service front.

Le (ou les) service back génère une API REST pour accéder (CRUD create, read, update, delete) aux informations des villes et des bâtiments.

Le service front propose à l'utilisateur une interface graphique en JavaFX. Il consomme l'API pour afficher les informations utiles à l'utilisateur.

## Milestones

### I. Modèle essentiel

Proposer une structure d'objets qui est capable de stocker l'information essentielle du projet. Ce modèle doit permettre de stocker des bâtiments de différentes villes avec leurs description, adresse, année de construction, architecte et liens vers les photos.

### II. Modèle intermédiaire

Proposer une structure d'objets dans laquelle il existe différentes catégories de bâtiments : château, maison, immeuble, ruines ...

### III. Modèle avancé

Un bâtiment peut être détaillé par l'ajout de spécificités concernant son architecture. Ces spécificités sont des éléments remarquables au niveau de la charpente, de la toiture, des fondations, des fenêtres par exemple

### IV. DAO et Base de données

Les informations de l'application sont persistées pour permettre au serveur de pouvoir redémarrer sans perte d'information.

### V. API essentielle



Les informations de Visit My Cities sont accessibles par une API REST. Celle ci permet d'accéder à la description de chaque bâtiment, de chaque ville, donne accès à tous les bâtiments d'une ville.

## VI. API avancée

L'API permet de voir les détails des spécificités liées aux bâtiments.

## VII. Interface essentielle

L'application permet à un visiteur de naviguer de ville en ville pour voir les différents bâtiments présent et d'en afficher les détails.

## VIII. Interface détaillée

L'application associe à la description d'un bâtiment un listing de ses différents éléments notables. S'il le souhaite, l'utilisateur faire afficher les détails de chacun de ces éléments.

## IX. Edition de bâtiments essentielle

Permettre à l'utilisateur d'ajouter ou de modifier des bâtiments.

## X. Edition de bâtiments intermédiaire

L'ajout ou modification de bâtiment n'est réalisable que par un utilisateur ayant été reconnu comme expert.


## XI. Edition de bâtiments avancée

Un utilisateur non expert peut proposer des modifications ou ajouts ; elles doivent être validées par un expert. Les experts sont notifiés de tout nouvel ajout lors de la connexion.

## XII. Filtrage des batiments

Permettre à l'utilisateur de filtrer les bâtiments qu'il affiche en fonction de leurs critères, de leur type, d'éléments notables inclus...

## XIII. Liste de voyage



L'application permet à un utilisateur de sélectionner les bâtiments qu'il souhaite visiter lors de son voyage dans la ville. Cette liste peut être à tout moment consultée ou modifiée par l'utilisateur.

#### XIV. Planificateur de trajet essentiel

A partir de la liste d'un utilisateur il est possible de générer automatiquement un trajet qui va comprendre l'ensemble des bâtiments à visiter dans l'ordre de la liste. Ce parcours peut être affiché dans un outil externe ou incorporé à l'application.

#### XV. Planificateur de trajet intermédiaire

L'utilisateur peut paramétrer le parcours inclure des contraintes dedans : temps de trajet maximum, moyen de transport, localisation actuelle pour le départ (manuelle ? automatique ?), ...

#### XVI. Planificateur de trajet avancé

Un algorithme est employé pour trouver un trajet théorique minime par rapport à la liste des bâtiments proposé. Le trajet propose donc de visiter les bâtiments en partant du plus proche de la localisation actuelle et en allant à chaque fois au plus proche suivant.

#### XVII. Prise en charge des fichiers images

Les images des bâtiments ne sont plus uniquement des url vers des images hébergées sur le web. Le service back prend en charge l'upload et le download d'images pour permettre aux utilisateurs d'afficher leurs propres photos.

#### XVIII. Prise en charge de la caméra

L'utilisateur peut utiliser sa caméra pour capturer l'image d'un bâtiment et l'associer à un bâtiment.

#### XIX. Connexion d'un utilisateur

L'application offre à l'utilisateur la possibilité de se connecter pour accéder à plus d'outils (ajout de bâtiment, liste de voyage)

#### XX. Inscription d'un utilisateur

L'application permet à un utilisateur de se créer un compte.

## XXI. Gestion des roles

L'application permet d'avoir plusieurs types de rôles : non connecté, visiteur et expert :

- Non connecté - accède à la liste des bâtiments
- Visiteur - a tous les droits du non connecté et a une liste de voyage et peut proposer des ajouts / modifications
- Expert - a tous les droits du visiteur et peut valider les propositions d'un visiteur, peut ajouter modifier un bâtiment sans validation, peut promouvoir un visiteur en expert.

## Ressources

Prendre en main les microservices avec Spring

[Construisez des Microservices](#)

Communication, sécurité, configuration de microservices

[Optimisez votre architecture Microservices](#)

Utiliser gitlab pour stocker votre projet et les taches à réaliser (clairement mieux que github sur ce point ci)

<https://openclassrooms.com/fr/courses/2035736-mettez-en-place-lintegration-et-la-livraison-continues-avec-la-demarche-devops/6182806-planifiez-votre-developpement>

Gerer les taches à part du repo git

<https://trello.com>

Des outils pour gerer son temps en période de télétravail

<https://toggl.com>

<https://clockify.me>