SMART CITY

Projet réalisé par  
Tano Iannetta  
Luana Martelli  
Wojciech Myszkorowski  
Camilo Pineda Serna  
Jérémie Zanone  
Loan Lassalle

Table des matières

[Introduction 3](#_Toc480623723)

[Cas d’utilisations 3](#_Toc480623724)

[Fonctionnalités principales 4](#_Toc480623725)

[Gestion, dans une base de données, de toutes les requêtes remontées par les citoyens 4](#_Toc480623726)

[Gestion d'un agenda 4](#_Toc480623727)

[Gestion des requêtes faites par les utilisateurs 4](#_Toc480623728)

[Filtrage des requêtes 4](#_Toc480623729)

[Ajout d’évènements de la ville 4](#_Toc480623730)

[Implémentation d'une carte interactive 4](#_Toc480623731)

[Generation d'un PDF 5](#_Toc480623732)

[Système de coordonnées 5](#_Toc480623733)

[Système WGS84 5](#_Toc480623734)

[Système OSM 6](#_Toc480623735)

[Conversion Système WGS84 en OSM 6](#_Toc480623736)

[Utilisation de tuiles 6](#_Toc480623737)

[Diagramme de classe 7](#_Toc480623738)

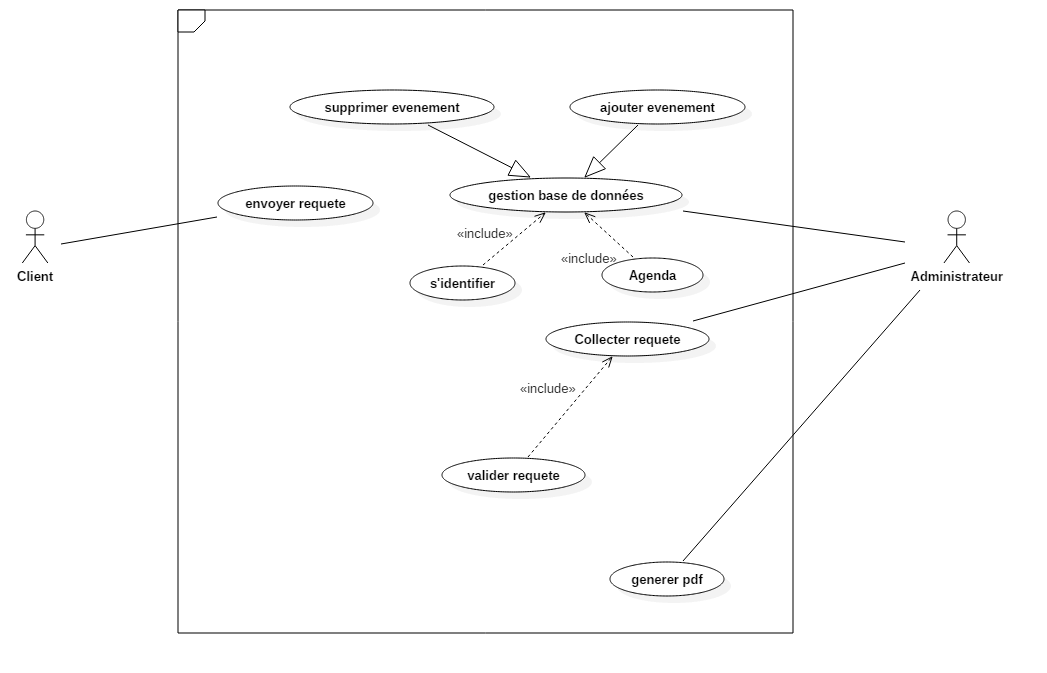
[Génération PDF 7](#_Toc480623739)

# Introduction

Dans le cadre de ce projet, nous voulons implémenter une application permettant à l'administration d'une ville (dans notre cas Lausanne) d'organiser les requêtes (de réparations, d'évènements, de projet, etc.) venant de ses citoyens. Les administrateurs de l'application pourront ajouter des évènements officiels.

Chaque requête devra être validée par un administrateur et ajoutée à une rubrique dédiée. Il sera ensuite possible de consulter ces différentes rubriques et de visualiser, au moyen d'une carte interactive, les endroits concernés. Pour une meilleure visibilité, chaque rubrique sera associée à un filtre, rendant ainsi plus aisée la lecture de la carte. Il sera aussi possible pour l'administrateur de donner des priorités aux évènements (haute ou à titre informatif) afin de traiter plus efficacement les requêtes. L'administrateur pourra aussi consulter la carte selon une date précise. Finalement, l'administrateur pourra générer des PDF contenant des informations relatives aux évènements et aux rubriques qu'il aura précédemment choisies.

# Cas d’utilisations



Dans la figure ci-dessus figure les cas d’utilisations.

# Fonctionnalités principales

## Gestion, dans une base de données, de toutes les requêtes remontées par les citoyens

* Les requêtes appartiendront à une rubrique (accidents, évènements, doléances, etc.)
* L'administrateur de l'application pourra gérer les rubriques.
* Un évènement aura une priorité donnée par l'administrateur.

## Gestion d'un agenda

* Pour toutes les requêtes, chacune aura une date de création et dans la mesure du possible une date de fin estimée.
* Organisation de la carte selon les filtres des rubriques et de la date (granularité au jour principalement).

## Gestion des requêtes faites par les utilisateurs

Les utilisateurs de l'application pourront effectuer plusieurs types de requêtes, classées selon les rubriques suivantes:

* Traffic
  + Accidents
  + Travaux
* Chantier
  + Constructions
  + Rénovations
* Culture
  + Manifestations
* Doléances

## Filtrage des requêtes

L'administrateur s'occupera de filtrer les requêtes faites par les utilisateurs. Pour certains comptes privilégiés, il n'y aura pas besoin de l'intervention de l'administrateur.

* Gestion des évènements (acceptation de requêtes, refus)
* Comptes privilégiés (par exemple, les accidents proposés par les TCS sont directement validés)

## Ajout d’évènements de la ville

Les utilisateurs pourront émettre une localisation d'une nouvelle manifestation qui sera approuvée ou non par l’administrateur.

## Implémentation d'une carte interactive

L'application comportera une carte interactive avec laquelle il sera possible d'interagir. Cette carte référencera à l'aide de "tags" les événements ayant lieu dans la ville

* Affichage interactif de la carte (zoom, déplacement manuel)
* Ajout de différentes icônes (pins) sur la carte pour localiser les différents types d'événements
* Filtre des événements par date et par rubriques

## Generation d'un PDF

* Génération d'un PDF selon une rubrique choisie
* Ces PDF seront principalement destinés aux départements reliés aux rubriques, afin de garder un historique des événements. On peut imaginer qu'une fois générés, ils seront envoyés aux personnes concernées.
* Un PDF sera organisé en deux parties
  + La première partie sera commune à tous les types de rubriques et contiendra les informations principales (nom de la rubrique, nom de l'événement, lieu, date, priorité et texte détaillant l'événement)
  + La deuxième partie sera présentée sous la forme de statistiques. Elles seront personnalisées selon la rubrique. Par exemple, s'il s'agit d'un événement relatif à des travaux, on affichera le temps moyen de la durée des chantiers. S'il s'agit en revanche de doléances, on affichera plutôt le nombre de commentaires qui ont été publiés à ce sujet, et ainsi de suite pour les autres rubriques.
* Si plusieurs filtres ont été sélectionnés, alors une option sera de générer automatiquement plusieurs PDF.

# Système de coordonnées

Dans notre projet nous utilisons une carte interactive. Afin de dessiner cette carte, nous avons utilisé un système de coordonnées. Nous avons utilisé 2 systèmes différents :

* Un système de coordonnée sphérique (3 dimensions, nommé WGS84, ce système est utilisé par les GPS notamment).
* Un système de coordonnée cartésien qui s’appellera ici OSM (système utilise par les principaux outils en lignes comme OpenStreepMap).

Le Système WGS84 est pratique et l’un des plus courant pour représenter les positions dans l’espace ainsi que calculer des positions. On utilise le système OSM car celui-ci est très utile pour l’affichage à l’écran.

## Système WGS84

Etant le système le plus utilisé de nos jours, nous avons décidé de l’utiliser afin de localiser nos différents évènements sur notre carte, c’est le système principal de notre projet.  
Un point est représenté par trois caractéristiques :

* **La longitude** est une [coordonnée géographique](https://fr.wikipedia.org/wiki/Coordonn%C3%A9e_g%C3%A9ographique) représentée par une valeur angulaire, expression du positionnement [est](https://fr.wikipedia.org/wiki/Est_(point_cardinal))-[ouest](https://fr.wikipedia.org/wiki/Ouest_(point_cardinal)) d'un point sur [Terre](https://fr.wikipedia.org/wiki/Terre).
* **La longitude** est une [coordonnée géographique](https://fr.wikipedia.org/wiki/Coordonn%C3%A9e_g%C3%A9ographique) représentée par une valeur angulaire, expression du positionnement [est](https://fr.wikipedia.org/wiki/Est_(point_cardinal))-[ouest](https://fr.wikipedia.org/wiki/Ouest_(point_cardinal)) d'un point sur [Terre](https://fr.wikipedia.org/wiki/Terre) (ou sur une autre [sphère](https://fr.wikipedia.org/wiki/Sph%C3%A8re)).
* **L'altitude** est l'élévation verticale d'un lieu ou d'un objet par rapport à un niveau de base. (dans ce projet ne tenant pas compte de cette composante car elle ne nous ait pas utile).

Le méridien de référence celui de Greenwich. La longitude est comprise entre -180o et + 180o les valeurs positives se trouvent à l’est et à l’ouest pour les valeurs négatives. La latitude se situe entre -90o et + 90o les valeurs positives se trouvant au nord et les valeurs négatives au sud. Dans ce projet nous n’avons pas tenu compte de l’altitude car ce paramètre ne n’affecte pas nos résultats d’affichages.

## Système OSM

Dans ce projet nous avons décidé d’utiliser les cartes de OpenStreepMap du fait qu’elles sont gratuites et que nous pouvons les utiliser de manière ouverte. OpenStreepMap offre un avantage considérable aussi, celui d’utiliser le système OSM qui est le système cartésien le plus utilisé actuellement ce qui nous a permis d’avoir de la facilité quand l’implémentation de ce système cartésien. Principalement ce système nous permet de visualiser la carte du monde avec différentes échelles (Zoom dans notre cas).  
Le zoom dans disponible est celui qui est compris dans un intervalle de [0 – 20] (le 20 est approximatif)  
Le zoom 0 correspond au zoom le plus éloigné et nous permet donc de voir la carte dans son entier. A l’inverse le zoom 20 est celui qui nous permet de nous rapprocher et de centraliser une zone très précise. Nous avons décidé d’utiliser une taille de 256 pixels pour chaque côté. A chaque niveau de zoom la carte est deux fois plus grande que l’image précédente. De manière général le niveau de zoom permet de calculer les dimensions de la manière suivante : 2x x 256 ( taille en pixel d’un coté) = 2x+8

## Conversion Système WGS84 en OSM

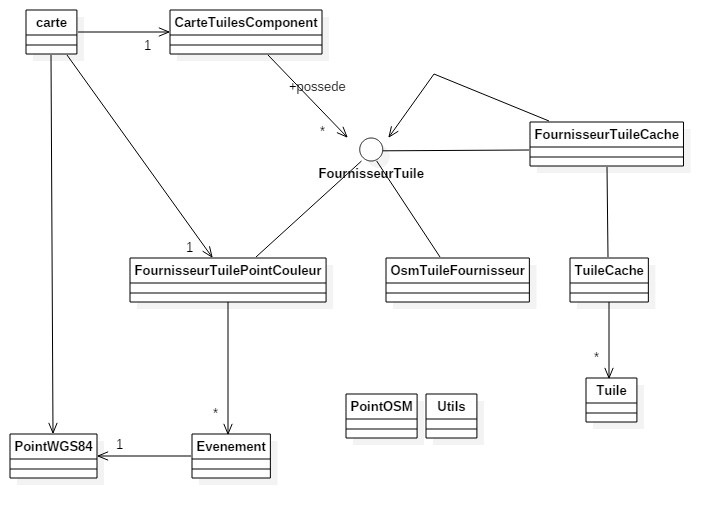
Pour passer d’un système à l’autre nous utilisons la projection cartographique, il existe une variété de projections permettant de passer d’un système à l’autre. Nous avons utilisé de Mercator. Pour chaque projection il existe plusieurs avantages et inconvénients, le désavantage de celui de Mercator est qu’il considère la terre comme étant une sphère parfaite. Ce qui n’est pas le cas dans la réalité.

# Utilisation de tuiles

Dans le cadre de notre projet, OpenStreepMap nous permet d’avoir un zoom d’un certain niveau dans notre cas un zoom de niveau 19 qui représente le zoom le plus proche. Si on prend la carte entière cela représente un nombre de pixels énorme de l’ordre 1815 pixels. Du cette quantité d’information, nous ne voulons utiliser qu’une infime partie. Afin de résoudre ce problèmes les gens utilisent des tuiles de petites tailles de 256 pixels de côté. Donc lors de l’initiation de la carte, on a seulement les tuiles visibles dans cette zone qui seront transmises au programme et assemblé au moment de l’affichage. Grâce à OpenStreetMap on identifie les tuiles avec 3 coordonnées le zoom, le x et le y.

Dans notre projet nous avons effectué plusieurs recherches afin de décider notre stratégie d’affichage de points colorié qui représentent les différents évènements. La création de tuile qui se superposent à notre carte nous permettent de tout simplement faire un calque sur notre carte et d’afficher les points par-dessus notre carte qui est une option plus facile à implémenter que le coloriage direct sur une carte.

## Diagramme de classe



Dans le schéma ci-dessus on a le schéma de la structure des classes utilisé pour construire notre carte.

# Génération PDF

Dans ce projet, il s’agit de garder une trace des données. Pour ce faire, nous avons décidé de donner la possibilité de générer un PDF contenant les informations relatives à la rubrique choisie. Ce document est composé de deux parties majeures. La première donne les informations importantes écrites des événements passés (comme le lieu et les dates). La deuxième partie est plus générale et comporte des statistiques.

Pour générer le PDF, nous avons choisi d’utiliser le kit de développement iText version 7. Ce support offre en effet plusieurs librairies qui permettent de créer facilement des PDF et laisse un grand degrés de liberté sur la gestion de la mise en page.

Pour la partie graphique, nous avons utilisé JfreeChart. Ce kit permet de créer toutes sortes de graphiques.

Les documents doivent pouvoir interagir avec la base de données pour récupérer toutes les informations nécessaires à sa création. Pour ce faire, nous avons créé une classe permettant de se connecter à la base de données et de lier les requêtes SQL avec des fonctions Java.

Finalement, un PDF est généré à partir d’un bouton sur l’interface graphique. Nous avons donc du lier les classes entre elles. De plus, si une erreur lors de la création du PDF est détectée, il faut pouvoir en informer l’utilisateur via l’interface graphique.