Smartcity

par

Tano Iannetta

Loan Lassalle

Luana Martelli

Wojciech Myszkorowski

Camilo Pineda

Jérémie Zanone

sous la direction du Professeur René Rentsch

Semestre de Printemps 2017

# Table des matières

[Table des matières 2](#_Toc483334392)

[1 Introduction 4](#_Toc483334393)

[1.1 But 4](#_Toc483334394)

[1.2 Objectifs 4](#_Toc483334395)

[1.3 Description 4](#_Toc483334396)

[1.4 Organisation 5](#_Toc483334397)

[1.5 Répartition des tâches 5](#_Toc483334398)

[2 Conception – Architecture 6](#_Toc483334399)

[2.1 Systèmes d’exploitation et outils logiciels 6](#_Toc483334400)

[2.2 Fonctionnalités principales 7](#_Toc483334401)

[2.2.1 Cas d’utilisations 7](#_Toc483334402)

[2.2.2 Gestion des requêtes dans une base de données 7](#_Toc483334403)

[2.2.3 Gestion d'un agenda 7](#_Toc483334404)

[2.2.4 Gestion des requêtes faites par les utilisateurs 8](#_Toc483334405)

[2.2.5 Filtrage des requêtes 8](#_Toc483334406)

[2.2.6 Ajout d'événements de la ville 8](#_Toc483334407)

[2.2.7 Implémentation d'une carte interactive 8](#_Toc483334408)

[2.2.8 Génération d'un PDF 9](#_Toc483334409)

[2.3 Technologies utilisées 9](#_Toc483334410)

[2.3.1 Java 9](#_Toc483334411)

[2.3.2 Git – GitHub 10](#_Toc483334412)

[2.3.3 Apache Maven 11](#_Toc483334413)

[2.3.4 Carte 12](#_Toc483334414)

[2.3.5 Base de données 14](#_Toc483334415)

[2.3.6 PDF 15](#_Toc483334416)

[2.3.7 GUI 16](#_Toc483334417)

[3 Implémentation 17](#_Toc483334418)

[3.1 Carte 17](#_Toc483334419)

[3.2 Base de données 18](#_Toc483334420)

[3.3 PDF 19](#_Toc483334421)

[3.4 GUI 20](#_Toc483334422)

[4 Tests de l’application 25](#_Toc483334423)

[4.1 Description générale de l’environnement de test 25](#_Toc483334424)

[4.1.1 Matériel 25](#_Toc483334425)

[4.1.2 Systèmes d’exploitation et outils logiciels 25](#_Toc483334426)

[4.2 Tests unitaires 26](#_Toc483334427)

[4.2.1 Carte 26](#_Toc483334428)

[4.2.2 Base de données 28](#_Toc483334429)

[4.2.3 PDF 29](#_Toc483334430)

[4.3 Tests d’intégrations 30](#_Toc483334431)

[4.4 Tests des cas d’utilisation 30](#_Toc483334432)

[5 Conclusion 31](#_Toc483334433)

[5.1 Etat des lieux 31](#_Toc483334434)

[5.1.1 Situation initiale 31](#_Toc483334435)

[5.1.2 Situation actuelle 31](#_Toc483334436)

[5.2 Problèmes rencontrés 31](#_Toc483334437)

[5.3 Propositions d’améliorations 31](#_Toc483334438)

[5.3.1 GUI 31](#_Toc483334439)

[6 Annexes 32](#_Toc483334440)

[6.1 Cahier des charges 32](#_Toc483334441)

[6.2 Journal de travail 33](#_Toc483334442)

[6.2.1 Tano Iannetta 33](#_Toc483334443)

[6.2.2 Loan Lassalle 34](#_Toc483334444)

[6.2.3 Luana Martelli 35](#_Toc483334445)

[6.2.4 Wojciech Myszkorowski 36](#_Toc483334446)

[6.2.5 Camilo Pineda 37](#_Toc483334447)

[6.2.6 Jérémie Zanone 38](#_Toc483334448)

[6.3 Planification 39](#_Toc483334449)

[6.4 Rapport de tests 39](#_Toc483334450)

[7 Sources – Bibliographies 40](#_Toc483334451)

[7.1 Personnes de références 40](#_Toc483334452)

[7.2 Ouvrages 40](#_Toc483334453)

[7.3 Sites internet 40](#_Toc483334454)

[7.4 Vidéos en ligne 40](#_Toc483334455)

# Introduction

## But

Aujourd’hui, la plupart des citoyens ont la possibilité de communiquer très rapidement avec des services privés ou publics. Cette opportunité leur permet de partager leurs avis et conseils afin d’améliorer les services.

Notre projet a consisté à l’élaboration d’une application nommée « SmartCity » qui assure l’archivage et la gestion de requêtes des différents acteurs de la société actuelle, l’affichage du lieu correspondant à une requête et la production de document PDF des données sélectionnées. Cette réalisation s’est faite dans le cadre du module « Projet » au Haute Ecole d’Ingénierie et de Gestion du Canton de Vaud (HEIG-VD).

Ce programme permet de gérer les requêtes venant de citoyens, de l’administration de la ville de Lausanne ou de « personnes de confiance », localiser de manière claire et précise le lieu correspondant à une requête et d’obtenir un résumé des différentes informations archivées.

Dans le cadre de ce rapport, nous exposerons les technologies qui nous ont permis de réaliser l’application. Nous décrirons les différentes fonctionnalités implémentées afin de mieux comprendre ce que permet notre application. Les problèmes rencontrés ainsi que les changements de conception seront aussi abordés afin d’expliquer nos choix quant au produit final.

Il est à noter que nous réalisons la partie administrative de l'application ; c'est-à-dire que notre projet sera utilisé par la ville et non pas par les citoyens de la ville.

## Objectifs

* Organiser les requêtes venant de différents acteurs identifiés (citoyens, administration de la ville de Lausanne, « personnes de confiance »)
* Afficher les lieux correspondant aux requêtes sur une carte interactive
* Produire des documents PDF pour fournir un résumé des informations archivées

## Description

Après plusieurs idées de projets, nous avons décidé de créer une application permettant à l’administration d’une ville (dans notre cas Lausanne) d'organiser des requêtes venant de ses citoyens, de « personnes de confiance » ou d’elle-même.

L’application s’articule autour de quatre actions principales.

La première consiste à la consultation d’événements présents au sein de la base de données. Ces événements peuvent être ajoutés par les citoyens au travers d’un site internet ou d’une application web (non réalisé dans le cadre du projet). Des entreprises ou des personnes bénéficiant de la confiance de l’administration de la ville obtiennent le statut de « personnes de confiance ». Elles peuvent, sans l’acceptation de l’administrateur, ajouter des évènements à la base de données. Ce droit est aussi accordé à l’administrateur de l’application.

La deuxième action est caractérisée par la gestion des requêtes stockées au sein de la base de données. Chaque requête créer par un citoyen lambda doit être validée par l’administrateur. Une fois validée, il est possible de consulter ces différentes rubriques et de visualiser, au moyen d'une carte interactive, les endroits concernés. Pour une meilleure visibilité, chaque rubrique est associée à un filtre, rendant plus aisée la lecture sur la carte.

La carte interactive permet à l’administrateur de se déplacer et d’agrandir le plan de la ville si besoin. La sélection d’une date lui offre le moyen d’affiner ces critères d’affichage. En ajoutant une priorité aux événements, l’administrateur sera notifié des requêtes possédant la priorité la plus haute dans le but de traiter le plus rapidement possible les requêtes.

La dernière action principale de l’applicatif est de générer des documents PDF contenant des informations relatives aux évènements et aux rubriques que l’administrateur aura précédemment choisies. Cela permettra un suivi « papier » pour un meilleur partage d’informations au sein de l’administration de la ville de Lausanne.

## Organisation

Chef de projet :

* Tano Iannetta : [tano.iannetta@heig-vd.ch](mailto:tano.iannetta@heig-vd.ch)

Chef en second :

* Wojciech Myszkorowski : [wojciech.myszkorowski@heig-vd.ch](mailto:wojciech.myszkorowski@heig-vd.ch)

Membres du projet:

* Loan Lassalle : [loan.lassalle@cpnv.ch](mailto:loan.lassalle@cpnv.ch)
* Luana Martelli : [luana.martelli@heig-vd.ch](mailto:luana.martelli@heig-vd.ch)
* Camilo Pineda : [camilo.pineda@heig-vd.ch](mailto:camilo.pineda@heig-vd.ch)
* Jeremie Zanone : [jeremie.zanone@heig-vd.ch](mailto:jeremie.zanone@heig-vd.ch)

## Répartition des tâches

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Phases/Membres | Tano Iannetta | Wojciech Myszkorowski | Loan Lassalle | Luana Martelli | Jeremie Zanone | Camilo Pineda |
| Dossier finale |  |  |  |  |  |  |
| Analyse |  |  |  |  |  |  |
| Conception |  |  |  |  |  |  |
| Réalisation |  |  |  |  |  |  |
| Administration |  |  |  |  |  |  |
| Tests |  |  |  |  |  |  |
| Documention de code |  |  |  |  |  |  |

# Conception – Architecture

## Systèmes d’exploitation et outils logiciels

Lors du choix des outils de conceptions, il était impératif de sélectionner des logiciels multiplateformes. Cette condition devait être remplie pour la bonne raison que les membres du projet ne possèdent pas le même système d’exploitation. La plupart des outils utilisés possèdent une licence qui a été contractée personnellement ou à travers les programmes de licences fournis par l’HEIG-VD.

* Microsoft Windows 10 Familiale 64 bits
* Microsoft Windows 10 Éducation 64 bits
* GNU/Linux Linux Mint 64 bits
* IntelliJ IDEA 64 Ultimate bits
* Java SE Development Kit 8
* Java SE Runtime Environment 8
* Microsoft Office Professional Plus 2016 64 bits
* MySQL Workbench 6.3
* Inkscape 0.48.5 64 bits
* StarUML 2.8.0

## Fonctionnalités principales

### Cas d’utilisations

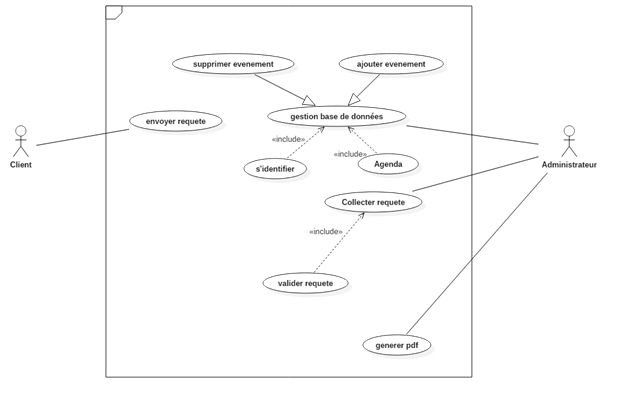


Diagramme des cas d’utilisations

L’administrateur est la personne qui utilisera l’applicatif. Cette personne aura la possibilité d’effectuer plusieurs actions. Elle peut tout d’abord valider des requêtes qui sont figurent dans une base de données. Cette personne peut aussi elle-même rajouter des événements dans la base de données alors modifier ceux y figurant déjà.

### Gestion des requêtes dans une base de données

### 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'événements 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 référence 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

### Génération 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.

## Technologies utilisées

### Java

Java est un langage de programmation informatique orienté objet, développé par Sun Microsystems. Nous avons décidé d’utiliser ce langage pour différentes raisons.

La première est que nous l’avons appris récemment en cours. Ce langage était encore frais dans nos esprits et son usage s’avère facilité. De plus, Javadoc permet de réaliser de la documentation de code claire et complète.

Java permet aussi d'avoir un très haut niveau d'abstraction par rapport à la machine. Sa portabilité contribue à l’exécution de programmes sur tout type de machine. Ainsi, elle offre une plus grande liberté aux utilisateurs.

Au niveau sécurité, Java est considéré comme un langage fiable, stable et sûr dù à l’utilisation d’une machine virtuelle Java. Un programme ne peut en aucun cas compromettre l’intégrité de cette dernière.

Ensuite, ce langage permet l’éxecution « simultanée » de plusieurs tâches pour obtenir de meilleures performances. Lors de l’utilisation d’interfaces graphiques, il est indispensable d’utiliser ce mécanisme.

Pour terminer, les deux principales raisons de notre choix ont été le grand nombre de librairies tierces et les améliorations constantes apportées à Java.

### Git – GitHub

Au sein de n’importe quel projet, il est primordial d’utiliser un système de versions de fichiers pour récupérer où archiver des documents. Ceci permet de suivre l’évolution des fichiers d’un projet, d’administrer les versions successives de plusieurs documents et de maintenir à jour une copie de ces fichiers à différents endroits où avec les différents membres du projet.

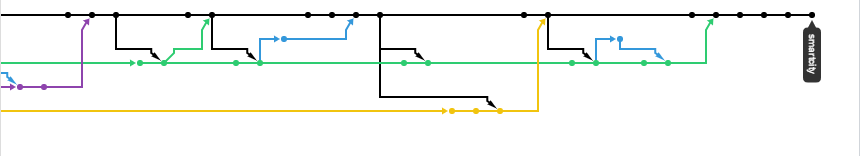
Dans ce projet, nous avons utilisé Git pour sa rapidité, sa taille extrêmement réduite, la gestion des branches et sa décentralisation. Multiplateforme, il a permis à chacun de travail dans son environnement préféré.



Les différentes « zones » de Git et les commandes correspondantes

En complément, nous avons utilisé la plateforme GitHub pour héberger et gérer le développement de logiciels. Elle nous a permis de mettre facilement et efficacement le travail de chacun en commun, d’obtenir une vue globale de l’évolution du projet et un échange d’informations plus rapides et à intervalle régulier.

L’atout majeur de GitHub est la possibilité d’accéder à une ressource en même temps que quelqu’un d’autre. Lorsque deux personnes ou plus travaillent sur la même ressource en même temps et décident d’envoyer au serveur distant, les modifications effectuées lèveront un conflit lors de la fusion des fichiers du projet. A ce moment-là, il faudra décider quelles parties garder de celles qui ont été modifiées.



Les branches du projet par rapport à une date de départ

### Apache Maven

Nous avons utilisé Maven pour la conception architecturale de notre projet.

Maven est un outil pour la gestion et l’automatisation de production Java en général et Java EE en particulier. L’objectif recherché est de produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication.

Maven utilise un paradigme connu sous le nom de Project Object Model (POM) afin de décrire un projet logiciel, ses dépendances avec des modules externes et l'ordre à suivre pour sa production. Il est livré avec un grand nombre de tâches prédéfinies, comme la compilation de code Java ou encore sa modularisation. Nous y trouvons entre autres :

* Le nom du projet
* Le numéro de version
* Les dossiers contenant le code source et les fichiers de ressources
* Les bibliothèques nécessaires à la compilation
* Les dépendances vers d’autres projets
* Les noms des contributeurs

Maven 2 insiste sur le principe de favoriser l'utilisation des conventions plutôt que de configurer son projet. Ainsi, si l'on respecte certaines conventions définies par Maven 2, il devient inutile de préciser certaines informations dans son *pom.xml*. Par exemple, Maven 2 préconise l'utilisation du répertoire *src/main/java* pour stocker les fichiers source du projet. En respectant ceci, il devient alors inutile de spécifier à Maven 2 où se trouvent les sources Java, ce qui allège d'autant l'écriture du *pom.xml*.

Avec ce simple *pom.xml* il devient possible de réaliser tout le processus de construction d'un projet avec Maven 2, à savoir :

* Gérer les fichiers de ressources
* Compiler les sources Java
* Compiler et exécuter les tests unitaires
* Créer le fichier JAR du projet
* Déployer le JAR

### Carte

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).

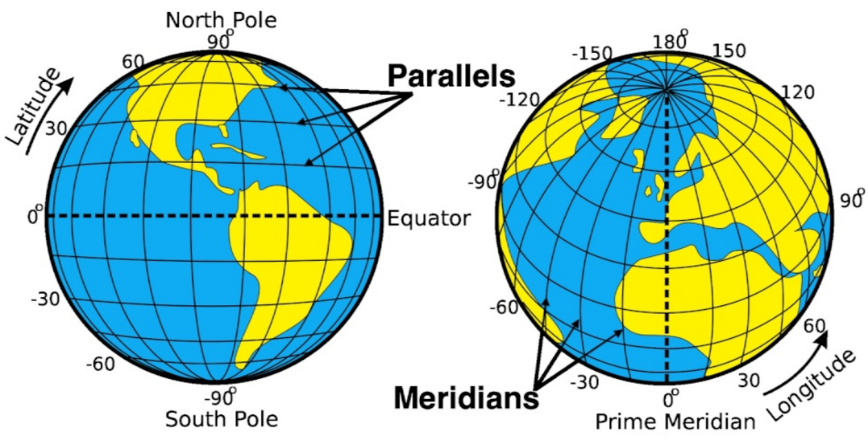
Nous avons décidé d'utiliser le Système WGS84 car celui est pratique pour plusieurs raisons. La première est que c'est un des plus courant pour représenter les positions dans l’espace. Deuxièmement il permet de calculer des positions sur des carte cartographié. Le 2eme système est utilisé pour l'affichage à l'écran grâce à sa facilité de d'adaptation aux système de pixels.

#### Système WGS84

Étant 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** et **la latitude** sont des coordonnées géographiques représentées par des valeurs angulaires, expression du positionnement est-ouest, respectivement nord-sud d'un point sur Terre (ou sur une autre sphère).
* **L'altitude** est l'élévation verticale d'un lieu ou d'un objet par rapport à un niveau de base. (Ce projet ne tient pas compte de cette composante car elle ne nous ait pas utile).



Représentation du système WGS84

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 de l’équateur. Dans ce projet, nous n’avons pas tenu compte de l’altitude car ce paramètre ne n’affecte pas nos résultats d’affichage.

#### 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 côté) = 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. Cela n'impacte pas les calculs de conversions. Les paramètres utilisés, sont choisis de telle manière que le point de coordonnées WGS 84 (0°,0°) se trouve au centre de la carte.

### Base de données

### PDF

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 degré 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.

### GUI

Pour l'interface graphique nous avons décidé d'utiliser la technologie swing, ce choix a été motivé par le fait que nous avions déjà travailler avec cette librairie par le passé lors d'un cours au semestre passé. Comme nous avions peu travailler avec des interfaces graphiques nous avons, en effet, préférer dans un terrain connu, surtout que notre application demandait une interface graphique relativement conséquente.

De plus, swing proposait tous les composants nécessaires à la réalisation de notre projet.

Il est en revange à noter que le design de swing n'est pas des plus élégant, mais nous l'avons jugé suffisant pour la réalisation de l'application dans laquelle nous avons mis l'importance sur les fonctionnalités. En outre, swing impose une syntaxe très lourde au code, et la relecture de ce dernier se retrouve fortement plus laborieuse. (Voir 5.3 Proposition d'amélioration.)

# Implémentation

## Carte

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.

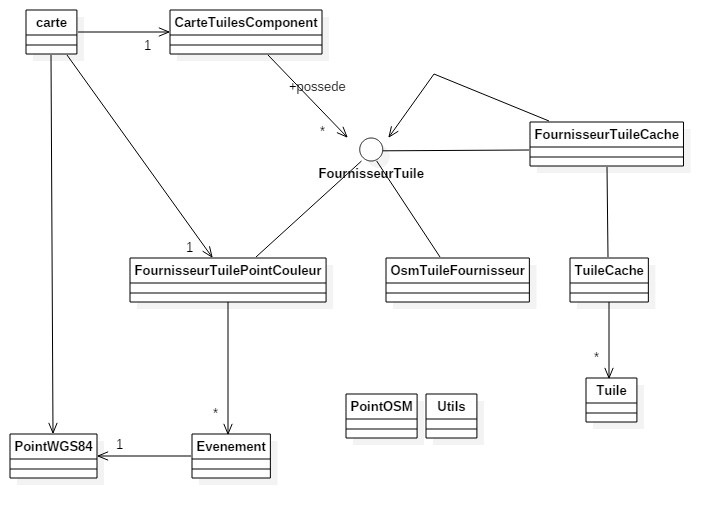


Schéma de la structure des classes utilisé pour construire notre carte

Nous avons une interface qui nous permet de redéfinir une méthode de calcul de tuile. L’utilisation des tuiles de openStreepMap nous a dirigé vers l’implémentation de ..

## Base de données

## PDF

Dans ce projet, il s’agit aussi 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.

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 dû 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.

Une classe principale, GenerateurPDF, est appelée lorsque l'utilisateur clique sur le bouton. Cette classe s'occupe de créer un document et de le remplir avec les éléments présents dans la base de données. Les données collectées ainsi servent à réaliser un graphique. À chaque création de PDF, un graphe en barres ou un graphe circulaire est généré de manière aléatoire.

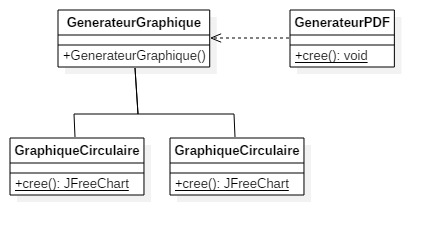


Schéma de la structure des classes pour la génération du PDF

## GUI

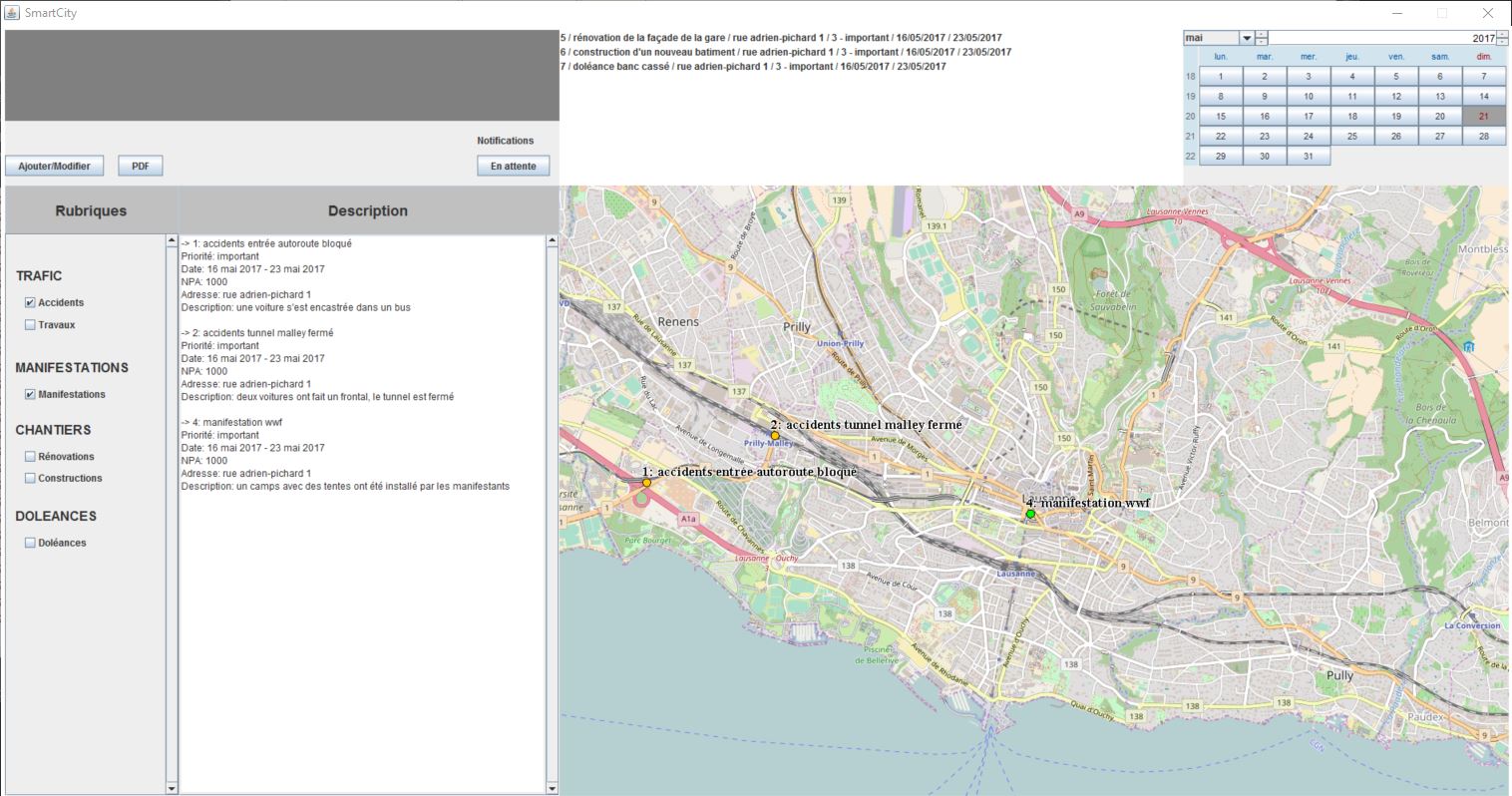
L'application est modélisée graphiquement par deux fenêtres, dont la seconde porte sur deux fonctionnalités distinctes mais similaires.

La première fenêtre est la fenêtre "principale" de l'application, c'est elle qui permet de voir l'affichage de la carte, et des événements sur cette dernière. Les check boxes et le calendrier agissent comme filtres sur les événements affichés.

En plus de voir les positions des événements sur la carte, nous voyons leur description respective permettant d'avoir toutes les informations sur les événements.

Nous voyons aussi une liste défilante afficher les événements en attente de validation par l'administrateur (l'utilisateur de notre application).

Cette fenêtre contient également les différents boutons offrants les autres fonctionnalités de notre application : Ajout et modification des événements, validation des événements en attente et génération d'un PDF.



10

5

1

7

6

9

8

3

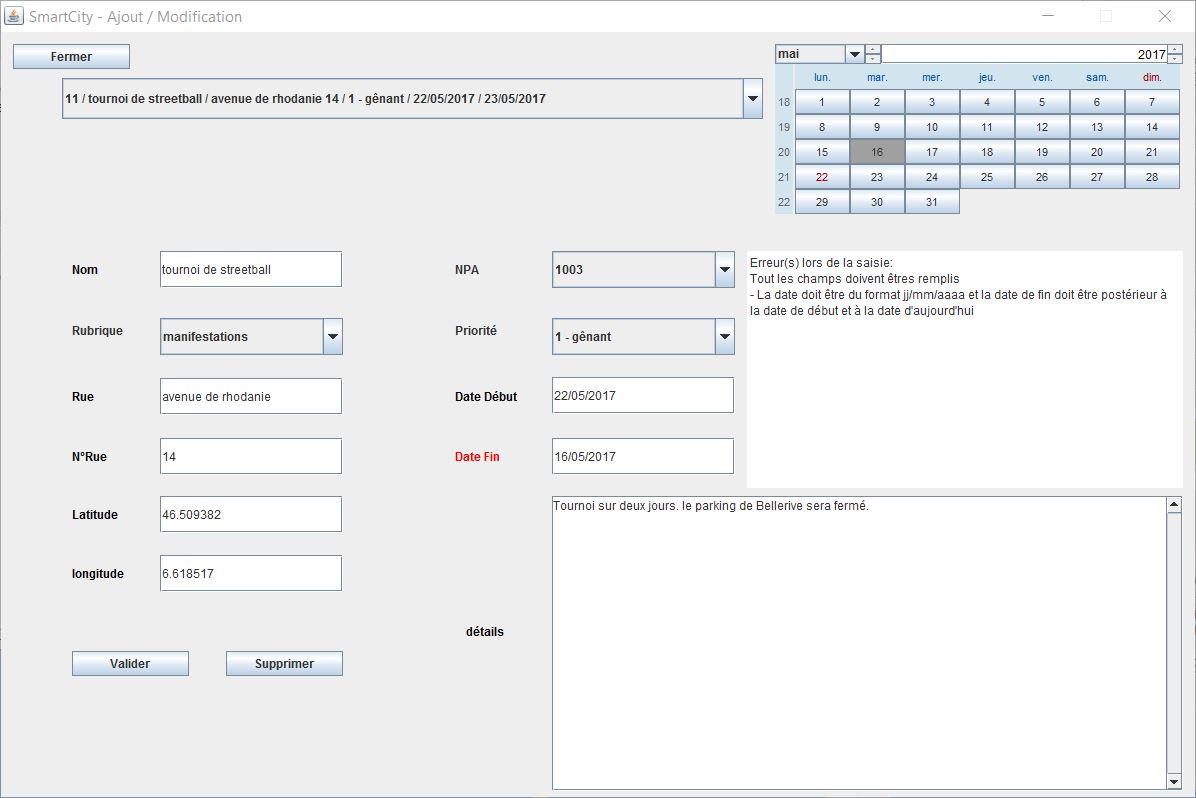
2

[TITRE IMAGE]

Description des compostant de la fenêtre principale :

|  |  |
| --- | --- |
| **Eléments / zone** | **Description** |
| 1. Logo | Représente le logo de l'application |
| 1. Bouton Ajouter/Modifier | Ce bouton ouvre sur la fenêtre d'ajout/modification d'événements. |
| 1. Bouton PDF | Ce bouton permet de générer un PDF comprenant les informations relatives à la rubrique choisie (grâce aux checkboxes). |
| 1. Notifications | Affiche le nombre d'événements en attente de validation |
| 1. Bouton en attente | Ouvre sur la fenêtre permettant de valider ou refuser en événement en attente. |
| 1. Checkboxes Rubriques | Les cases cochées agissent comme filtre sur les événements à afficher sur la carte interactive. |
| 1. Description | Affiche les descriptions complètes des événements filtrés |
| 1. Liste événements en attente | Liste déroulantes affichant les événements en attente de validation. |
| 1. Calendrier | Agit comme filtre par la date sur les événements à afficher |
| 1. Carte | Carte interactive affichant la position des événements filtrés |

La deuxième fenêtre survient dans deux contextes différents. Le premier est lorsque l'on veut ajouter ou modifier un événement de la base de données. La distinction entre l'ajout ou la modification est faite par la liste déroulante. Cette dernière contient les événements actifs dans la base de données (ceux dont la date de fin n'est pas passée) et une option d'ajout d'un nouvel événement. Lorsque l'on choisit un événement déjà présent dans la base de données avec la liste déroulante, les champs sont remplis avec les attributs de l'événements sélectionné. On peut alors modifier la valeur de ces champs pour modifier l'événement sélectionné.

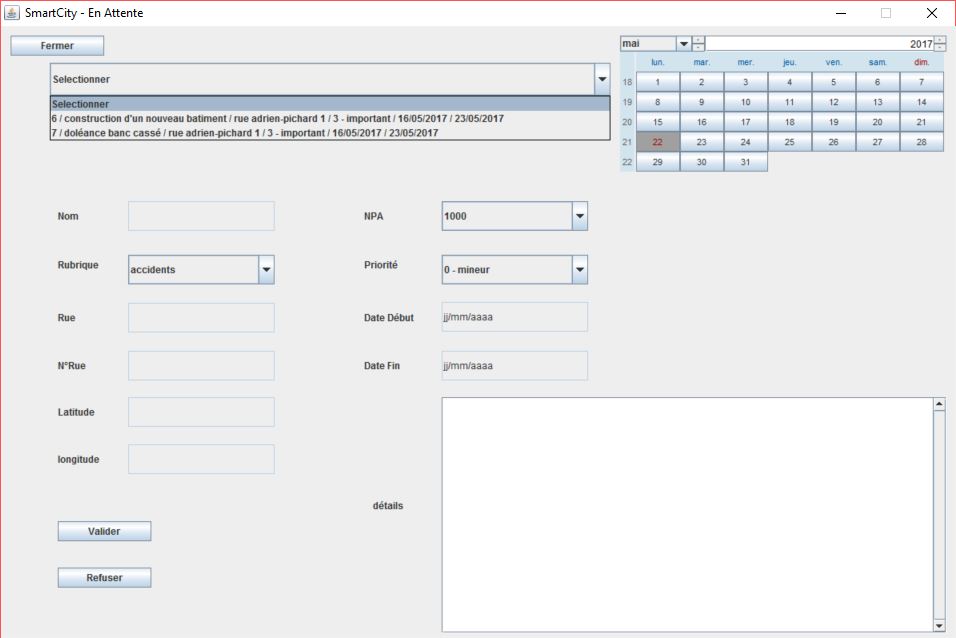


[TITRE IMAGE]

Description des composants de la fenêtre d'ajout/modification :

|  |  |
| --- | --- |
| **Eléments / zone** | **Description** |
| 1. Liste déroulante | Permet de choisir si l'on veut ajouter un événement ou modifier un élément déjà présent dans la base de données |
| 1. Champs | Champs éditables pour donner les attributs d'un événement |
| 1. Calendrier | Permet de remplir les champs de date de début et de date de fin en interagissant avec les boutons du calendrier. |
| 1. Notification d'erreurs | Indique à l'utilisateur quel champs est mal complété et la forme attendue |
| 1. Bouton fermer | Ferme la fenêtre courante. |
| 1. Bouton valider | Contrôle les données saisies, et si elles sont valides, ajoute l'événement à la base de données, ou met à jour l'événement si l'utilisateur avait choisi un événement déjà présent dans la base de données. |
| 1. Bouton Supprimer | Supprime l'événement sélectionné de la base de données |

Le second contexte est la validation d'événements en attente. Dans ce cas, nous pouvons valider ou refuser un événement. La liste déroulante permet de sélectionner un des événements en attente, la sélection de ce dernier rempliera les champs de la fenêtre. Nous pouvons ensuite valider l'événement en effectuant ou non des modifications.



[TITRE IMAGE]

Les boutons suivants différent par leur fonctionnalité du contexte d'ajout/modification d'événements.

|  |  |
| --- | --- |
| **Eléments / zone** | **Description** |
| 1. Liste déroulante | Permet de sélectionner un événement en attente de confirmation. |
| 1. Bouton valider | Modifie l'événement en fonction du contenu des champs (s'ils diffèrent) et valide l'événement |
| 1. Bouton refuser | L'événement n'est pas validé. |

Ces deux contextes implémentent une validation du contenu des champs. Si une valeur n'est pas conforme aux espérances, le label de la rubrique est affiché en rouge et une note est affichée et explique ce qui est attendu dans le champs mal complété.

# Tests de l’application

## Description générale de l’environnement de test

Durant notre projet nous avons procédé à différents tests lors du développement de l'application. Nous avons tâché d'isoler au mieux les différentes parties telles que la génération de PDF et la carte par exemple afin de pouvoir les tester de manière isolée et de déceler les choses qui ne fonctionnaient pas comme nous voulions. De manière générale, nous nous sommes fixé des objectifs de fonctionnalité qui devait être implémenté et nous avons testé dans un premier temps que celles-ci soient fonctionnels.

Au niveau de la sécurité logiciel de l’application, nous nous sommes concentrés sur les contrôles d’entrer de l’utilisateur. La base de données et la communication avec l’applicatif font partie des éléments critiques à protéger.

Deux machines virtuelles étaient dédiées aux tests de validation. Elles tournaient sous Microsoft Windows 7 SP1 64 bits et Microsoft Windows 10 Familiale 64 bits. Elles ont permis de tester l’application dans des environnements très utilisé aujourd’hui.

Des tests unitaires et d’intégrations ont été effectués durant le développement de l’application sur la machine possédant Microsoft Windows 7. La base de données était stockée en local. Les données utilisées lors des tests étaient des données fictives.

Les conditions exactes des tests unitaires et d’intégrations du projet étaient :

* Microsoft Windows 7 64 bits
* Une connexion au réseau Internet et Intranet de l’HEIG-VD d’Yverdon-les-Bains
* L’archive JAR du projet située à la racine de son disque dur système

Une campagne de validation des cas d’utilisation a été effectuée sur les deux machines virtuelles par deux personnes extérieures au projet, à la fin du développement de l’applicatif. Ceci nous a permis d’obtenir des avis objectifs sur la correspondance entre l’applicatif et les cas d’utilisations annoncées. Ainsi nous avons pu valider le bon fonctionnement de l’application.

### Matériel

* Deux machines virtuelles de test
  + Intel® Core™ I7 4770 3.40GHz
  + 1 coeur physique, 2 coeurs logiques
  + 2 Go de RAM
  + Connexion réseau : Host-only
  + Taille du disque dur : 40.0 GB

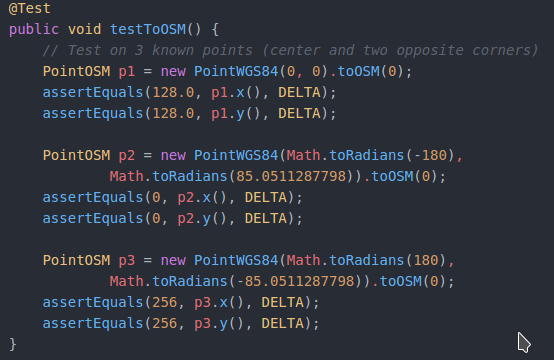
### Systèmes d’exploitation et outils logiciels

* Microsoft Windows 7 Entreprise 64 bits SP1
* Microsoft Windows 10 Familiale 64 bits

## Tests unitaires

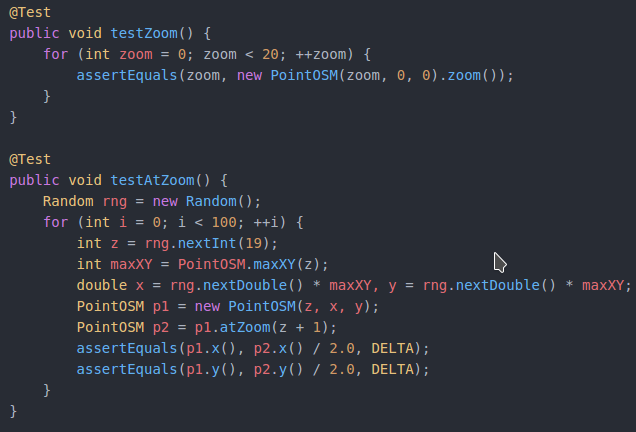
### Carte

Afin de tester certaines méthodes nous avons utilisé des tests unitaires à l'aide de Junit qui permet d'insérer des test dans l'application et de référence des méthodes présentes dans le corps du projet de vérifier qu'elles fassent ceux à quoi on s'attend. Dans le cas de la carte il était par exemple question de tester la validité des conversions entre les ponts wgs84 en points OSM. Nous avons pour cela utiliser des tests unitaires.



Test unitaire pour tester la conversion des point WGS84 en OSM

Nous avons aussi effectué des tests sur le zoom qui ne peut pas dépasser certaines valeurs.



[TITRE IMAGE]

### Base de données

### PDF

## Tests d’intégrations

## Tests des cas d’utilisation

# Conclusion

## Etat des lieux

### Situation initiale

### Situation actuelle

#### Fonctionnalités non implémentées

## Problèmes rencontrés

Nous avons rencontré plusieurs problèmes auxquels nous n’étions pas préparés ou que nous avions sous-estimé la charge de travail. Lors de l'élaboration initiale du projet, nous avions conscience de la difficulté de certaines tâches, mais nous ne savions pas précisément quelles technologies nous allions utiliser pour développer l’application et ces différents composants.

Tout d’abord, la mise en place de la base de données fut beaucoup plus longue que prévu. Cela a retardé l’accès et l’intégration des données au projet.

Composant phare de notre application, la carte interactive affichait des performances en deçà de nos attentes. Il a fallu trouver un moyen de pouvoir charger et afficher rapidement les tuiles de la carte. Après avoir effectué des recherches, nous avons compris qu’il était possible de mettre en cache les tuiles. Ce processus permet de les afficher plus rapidement au sein de l’application.

Une autre difficulté rencontrée fut de rassembler les principaux composants (carte, base de données et génération de document PDF). Au début de l’élaboration des composants, nous avons décidé de partir sur des sous-projets Java sans utiliser Apache Maven. Mais lors de l’intégration des composants, nous avons eu des difficultés en raison de l’utilisation de différentes librairies pour chacun des sous-projet. Pour remédier à ce problème, nous avons décidé de convertir les composant en sous-projet Maven afin d’avoir plus de facilité lors de l’intégration.

## Propositions d’améliorations

## 5.3.1 GUI

Lors du choix de la librairie de l'interface graphique, nous ne connaissions que JavaSwing. Cette dernière suffisait à l'implémentation de notre application. Mais nous avons appris l'existence de JavaFX, qui à l'instar de Swing permet d'implémenter une interface graphique. Toutefois, JavaFX s'organise différemment et permet d'avoir un code beaucoup plus organisé et facile à lire. De plus, le design est plus agréable.

Une amélioration possible de notre application serait de passer de Swing à JavaFX. Nous n'avons pas pu effectuer cette modification dans les délais étant donné qu'une grosse partie de notre application concerne la partie graphique et que cela représentait une trop grosse charge de travail pour être fini à temps.

# Annexes

## Cahier des charges

## Journal de travail

### Tano Iannetta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Semaine | Activités | | | |
| Tâche planifiée | Résultat du travail effectué | Heures prévues/  effectives |
| Semaine 1 | Recherche de sujet et discussion de groupe | Choix du 1er sujet | 3/3 |
| Semaine 2 | Mise en place github  Rédaction readme  Canevas | Travaille sur le cahier des charges du 1er sujet  Recherche d'un 2eme sujet | 6/8 |
| Semaine 3 | Choix 2eme sujet  Début interface graphique | Choix du 2eme sujet  Cahier des charges | 9/6 |
| Semaine 4 | Avance interface graphique | Spécification de l’interface graphique des deux fenêtres de l'application (avances générale) | 5/5 |
| Semaine 5 | Avance de la fenêtre principale de l'interface graphique, Maintenance github | Spécification et implémentation de l’interface graphique des deux fenêtres de l’application (avances générale) | 9/8 |
| Semaine 6 | Finalisation de la fenêtre principale | Implémentation de la fenêtre de modification des événements et du remplissage des champs | 7/9 |
| Semaine 7 | Communication avec la base de données, Apprentissage Java pour la communication avec une base de données | Contrôle de saisie de la fenêtre de modification + intégration de la carte dans la fenêtre principale | 8/7 |
| Semaine 8 | Continuation de la communication entre base de données et fenêtre principale | Préparation de la présentation  Tests et corrections de la fenêtre de modification  (valeurs fictives, sans base de données) | 4/4 |
| Semaine 9 | Gestion interface graphique avec la carte interactive | Intégration de la base de données avec l'interface graphique (ajout et suppression d'événements.) | 8/7 |
| Semaine 10 | Implémentation de la fenêtre d'ajout/modification des événements | Intégration de la base de données avec l'interface graphique (ajout et suppression d'événements.) | 4/7 |
| Semaine 11 | Fusion des fonctionnalités | Adaptation du code et corrections de divers bugs de la base de données et l'interface graphique  Implémentation de la fonctionnalité de modifier un événement | 5/8 |
| Semaine 12 | Merge des fonctionnalités | Tests et corrections de la fenêtre de modification avec la base de données | 5/5 |
| Semaine 13 | Tests de l'application | Finalisation du code de l'interface graphique, Tests de l'application | 5/4 |

### Loan Lassalle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Semaine | Activités | | | |
| Tâche planifiée | Résultat du travail effectué | Heures prévues/  effectives |
| Semaine 1 | Constitution des groupes, choix du sujet et discussions | Nomination du chef de projet et du second, proposition de projet : coffre-fort numérique | 3/4 |
| Semaine 2 | Commencement du cahier des charges, feedback du professeur | Début du cahier des charges, recherche d’un nouveau sujet en raison du refus du précédent sujet, proposition de projet : gestionnaire de requêtes citoyennes, rédaction du cahier des charges du nouveau projet | 3/6 |
| Semaine 3 | Cahier des charges du projet avec planning | Finalisation du cahier des charges et du planning | 5.5/5.5 |
| Semaine 4 | Conception de la base de données | Conception de la base de données, recherche des technologies adaptées au projet pour la liaison avec la base de données | 6/8 |
| Semaine 5 | Conception de la base de données | Choix de la technologie utilisé : HIBERNATE, compréhension du fonctionne et de son implémentation | 6/10 |
| Semaine 6 | Réalisation de la base de données, implémentation de la liaison avec la base de données | Modification et mise en place de la base de données, implémentation de la traduction de code Java en requête SQL | 6/10 |
| Semaine 7 | Réalisation de la base de données | Finalisation de l’implémentation de fonctionnalité | 7/8 |
| Semaine 8 | Intégration de la base de données avec l’interface graphique | Mise en commun avec la carte, simplification de l’implémentation | 7/7 |
| Semaine 9 | Intégration de la base de données avec l’interface graphique | Correction de la structure de la base de données (trigger), intégration de la liaison de la base de données avec l’interface graphique | 7/7 |
| Semaine 10 | Intégration de la base de données avec l’interface graphique | Mise en place de l’ajout d’évènements au sein de l’interface graphique, simplification de fonctionnalités | 7/7 |
| Semaine 11 | Intégration de la base de données avec l’interface graphique, rédaction du rapport final | Mise en place de la modification d’évènements au sein de l’interface graphique, mise en place du rapport, début de rédaction | 5.5/7 |
| Semaine 12 | Rédaction du rapport final | Rédaction du rapport final, finalisation des commentaires dans le code | 5.5/7 |
| Semaine 13 | Rédaction du rapport final | Finalisation du rapport et journal de bord | 5.5/7 |

### Luana Martelli

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Semaine | Activités | | | |
| Tâche planifiée | Résultat du travail effectué | Heures prévues/  effectives |
| Semaine 1 | Constitution des groupes, choix du premier sujet et discussions | C.f. tâche planifiée | 4 / 4 |
| Semaine 2 | Cahier des charges, recherche d’un 2ième sujet | C.f. tâche planifiée, ébauche d'analyse du PDF | 6 / 6 |
| Semaine 3 | Cahier des charges du 2ième projet avec planning | C.f tâche planifiée, ébauche d'analyse des statistiques | 6 / 6 |
| Semaine 4 | Recherches / analyse génération de PDF | Recherche de libraires / JDK, première ébauche de PDF | 6 / 6 |
| Semaine 5 | Recherches / analyse génération de graphiques | Ajout des statistiques au PDF | 5 / 4 |
| Semaine 6 | Implémentations | Données écrites en brut, commentaires, nettoyage du code | 6 / 6.5 |
| Semaine 7 | Implémentations | Structure du code, continuation du PDF | 6 / 4 |
| Semaine 8 | Implémentations | Structure du code, continuation du PDF | 5.5 / 5 |
| Semaine 9 | Implémentation | Début de la connexion avec la base de données | 5.5 / 5 |
| Semaine 10 | Liaison du PDF avec la base de données | Connexion avec la base de données / peuplement | 6 / 6 |
| Semaine 11 | Liaison du PDF avec la base de données | Fin de la connexion avec la base de données / début de la connexion avec l'interface graphique | 6 / 10 |
| Semaine 12 | Rédaction du rapport final / Connexion PDF-interface graphique | Connexion avec l'interface graphique, tests | 6 / 10 |
| Semaine 13 | Rédaction du rapport final | Rédaction du rapport | 6 / 5 |

### Wojciech Myszkorowski

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Semaine | Activités | | | |
| Tâche planifiée | Résultat du travail effectué | Heures prévues/  effectives |
| Semaine 1 | Constitution des groupes, choix du premier sujet et discussions | c.f: tache planifié | 3/3 |
| Semaine 2 | Cahier des charges, recherche d’un 2ième sujet | c.f: tache planifié | 5/6 |
| Semaine 3 | Cahier des charges du 2ième projet avec planning | c.f: tache planifié | 5/7 |
| Semaine 4 | Mise au point implémentation de la carte | Conception théorique des points OSM 1er jet du diagramme de classe de la carte | 5/6 |
| Semaine 5 | Recherches sur les systèmes de points wgs84 et OSM  Codage des premières classes de la carte | Mise en place de classe Utils et methodes de conversion des points. mise en place de tests pour vérifier les calculs de conversion | 5/6 |
| Semaine 6 | Implémentation de la carte  Avancement code sur les tuiles | Implémentation des classes présentes dans le diagramme de classe pour la carte | 5/6 |
| Semaine 7 | Implémentation de la carte | Ajout de la carte à l'application | 5/3 |
| Semaine 8 | Implémentation des points d'affichage sur la carte | Résolution des bugs sur la carte en ce qui concerne le zoom et le déplacement long | 7/7 |
| Semaine 9 | Résolution des problèmes d'ajouts de la carte à l'application principale | Résolution des bugs sur la carte en ce qui concerne la mise en cache des tuiles | 7/7 |
| Semaine 10 | Résolutions des bugs sur la carte | Rajout de commentaires dans les classes de la carte et mise au propre du code | 7/7 |
| Semaine 11 | Codages des dernières fonctionnalités de l'application | Rédaction du rapport final  Implémentation des notifications sur la GUI | 7/7 |
| Semaine 12 | Rédaction du rapport final  Et tests finaux sur l'application | Rédaction du rapport final  Implémentation de la mise à jour des notifications des notifications dans la Base de données | 7/5 |
| Semaine 13 | Finalisation rapport et relecture | Rédaction du rapport final  Tests finaux pour l'application | 5/7 |

### Camilo Pineda

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Semaine | Activités | | | |
| Tâche planifiée | Résultat du travail effectué | Heures prévues/  effectives |
| Semaine 1 | Constitution des groupes, choix du premier sujet et discussions | Cf : tâche planifiée | 4/3 |
| Semaine 2 | Cahier des charges, recherche d’un 2ième sujet | Cf : tâche planifiée | 6/5 |
| Semaine 3 | Cahier des charges du 2ième projet avec planning | Cf : tâche planifiée | 4/4 |
| Semaine 4 | PDF et génération de fichiers | Conception base de données | 4/3 |
| Semaine 5 | PDF et Conception de la base de données | PDF première partie, titre, infos | 6/6 |
| Semaine 6 | GUI Réalisation de la base de données | PDF début seconde partie, statistiques | 6/5 |
| Semaine 7 | Rapport, PDF |  | 6/5 |
| Semaine 8 | Documentation, GUI | Assistance GUI | 6/5 |
| Semaine 9 | PDF tests, GUI | rapport | 6/5 |
| Semaine 10 | Mise en commun code, documentation | Assistance GUI | 6/5 |
| Semaine 11 | Documentation rapport, mise en commun | Rédaction rapport et manuel utilisateur | 6/5 |
| Semaine 12 | Tests, debug | Cf : tâche planifiée + documentation : manuel utilisateur | 6/8 |
| Semaine 13 | Rapport et documentation, tests | Cf : tâche planifiée | 6/ |

### Jérémie Zanone

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Semaine | Activités | | | |
| Tâche planifiée | Résultat du travail effectué | Heures prévues/  effectives |
| Semaine 1 | Constitution des groupes, choix du premier sujet et discussions | Cf: tâche planifiée | 2.5/3 |
| Semaine 2 | Cahier des charges, recherche d’un 2ième sujet | Cf: tâche planifiée | 5.5/5.5 |
| Semaine 3 | Cahier des charges du 2ième projet avec planning | Cf: tâche planifiée | 5.5/5.5 |
| Semaine 4 | Conception de la carte et implémentation de point pour la carte | Conception de la carte et implémentation de points de coordonnées pour les événements sur la carte ainsi qu'implémentation des événements | 6.5/7 |
| Semaine 5 | Création des tuiles pour la carte | Création des fournisseurs de tuiles et de l'interface graphique pour les tuiles de la carte | 5.5/6 |
| Semaine 6 | Création de l'interface graphique contenant les tuiles | Continuer l'interface graphique du carteTuilesComponent + tests d’affichages et résolutions de beug | 5.5/5.5 |
| Semaine 7 | Ajouter un pin de couleur sur la carte pour chaque événement | Ajout des informations et pin selon une liste d'événements sur la carte + ajouter la carte à la fenêtre principale | 5.5/8 |
| Semaine 8 | Avancer le code pour la présentation intermédiaire | Ajout de l'interaction de la carte avec la souris | 5.5/6 |
| Semaine 9 | Ajout de l'interaction sur la carte | Poursuite de l'interaction carte et ajout d'une classe pour garder les tuiles utilisées dans un cache (ArrayList) | 6/9 |
| Semaine 10 | Apprendre comment récupérer les informations depuis la base de données | Ajouter les points des événements sur la carte selon les checkboxes et la date sélectionnées | 5/7 |
| Semaine 11 | Linker la base de données avec la carte | Ajouter les descriptions des événements selon les checkboxes et la date sélectionnées | 7/6 |
| Semaine 12 | Rédaction du rapport final | Cf: tâche planifiée + amélioration du code | 8/8 |
| Semaine 13 | Rédaction du rapport final | Cf: tâche planifiée + commenter le code | 8/9 |

## Planification

Comparaison entre ce qui a été prévu et ce qui a été fait

## Rapport de tests

# Sources – Bibliographies

## Personnes de références

* Thibaud Besseau - HIBERNATE

## Ouvrages

* NOM, prénom de l’auteur. Titre de l'encyclopédie. Mention d’édition, ville, éditeur, collection, année, nombre de volumes ou tomes s’il y a lieu ou nombre de pages
* SAVITCH Walter. Absolute Java. 5ième édition, San Diego, Pearson, 2013, 1260 pages
* DELANNOY, Claude. Programmer en Java. 9ième édition, Saint-Germain, Eyrolles, 2015, 916 pages

## Sites internet

Les sites internet ci-dessous ont tous été consulté du 20.02.2017 au 30.05.2017.

* Nom, prénom de l’auteur (ou nom de l’organisme) : Adresse URL
* Developpez : <http://www.developpez.com/>
* iText : <http://itextpdf.com/>
* JBoss HIBERNATE : <https://docs.jboss.org/hibernate/stable/core.old/reference/fr/html/>
* JFreeChart : <http://www.jfree.org/jfreechart/>
* Microsoft : <http://blogs.msdn.com/>
* Microsoft : <http://www.labo-microsoft.org/>
* Microsoft : <https://msdn.microsoft.com/>
* Microsoft : <http://support.microsoft.com/>
* Stackoverflow : <http://stackoverflow.com/>
* Wikipédia : <http://wikipedia.org/>
* Youtube : <https://www.youtube.com/>

## Vidéos en ligne

* Auteur ou Origine. Titre de la vidéo ou description. [Indicateur Internet]. <Adresse URL >. (Consulté le JJ.MM.AAA).
* edureka!. Hibernate Tutorial - 1 | Hibernate Tutorial for Beginners - 1 | Edureka. [Tutoriel en ligne]. <https://www.youtube.com/watch?v=iwMW\_SZPGnY>. (Consulté le 13 mars 2017).
* edureka!. Hibernate Tutorial | Hibernate Tutorial - 2 | Hibernate Tutorial for Beginners - 2 | Edureka. [Tutoriel en ligne]. <https://www.youtube.com/watch?v=VAr\_IikcYfE>. (Consulté le 13 mars 2017).
* edureka!. Hibernate - The Ultimate ORM Framework | Edureka. [Tutoriel en ligne]. <https://www.youtube.com/watch?v=xo\_HGo812DQ>. (Consulté le 13 mars 2017).