Master 1 Informatique
Spécialité AIGLE

TER Master 1 Informatique

Que pense-t-on de Montpellier Agglomération ?

Responsables :

Eric Kergosien
Sandra Bringay
Maguelonne Teisseire
Mathieu Roche

Date de soutenance …
Membres du jury…

Amayas ABBOUTE
Gilles ENTRINGER
Franck PETITDEMANGE
Chaymae REGRAGUI

SOMMAIRE

PARTIE1 CAHIER DE CHARGE

1.1 Définition des besoins

1.1.1 Contexte général

Le projet SENTERRITOIRE a pour vocation la réalisation d'un site web dynamique s'appuyant sur une application existante : "TERRIDOCViewer". L'un des objectifs de ce projet est l'enrichissement de cette dernière, notamment en y introduisant de nouvelles composantes qui permettraient d'amplifier les descripteurs, géo-spatiaux et temporels, des informations liées à l'aménagement du territoire "Montpellier Agglomération ", pour doter cette application d'une accessibilité attractive au niveau de la recherche et de la visualisation.

Ces informations seront extraites automatiquement à partir d'un fonds documentaire constitué de documents d'actualités sous forme de textes ; sélectionnés et analysés par des experts géographes afin d'évaluer les opinions des acteurs (exemple : Auteur d'un document choisi) quant à ces données.

L'objectif de ce cahier des charges est d'expliquer les grandes lignes du projet :

Nous préciserons, dans un premier temps, les différents besoins indispensables à la réalisation de l'application et ainsi, au rendu espéré. Nous examinerons attentivement ces derniers lors de la définition des aspects fonctionnels concernant le projet ; en avant dernière partie de ce cahier des charges.

Ensuite, nous évoquerons les méthodes logicielles utilisées, et concevrons une architecture appropriée au travail demandé et adaptable à TERRIDOCViewer.

Enfin, nous terminerons sur les aspects techniques et ergonomiques en étudiant les points sur lesquels nous nous basons et ceux sur lesquels nous cherchons à être novateurs et soigneux. Cela grâce à des veilles et des études comparatives des différentes technologies optées et également en proposant une ergonomie que nous jugeons captivante.

1.1.2 Composante spatiale

L'intégration d'une composante spatiale dans l'application devra permettre de représenter et de mettre en valeur les différentes opinions dans un espace géographique sélectionné par l'utilisateur. La composante graphique devra être représentée sous la forme d'une carte de type Google Maps.

L'intégration de fonds de cartes "open source" peut s'avérer nécessaire. Selon les conditions d'utilisation de l'API GoogleMaps, l'éditeur de l'application doit acquérir une license payante à partir du moment que l'application génère plus que 25.000 cartes par jour.

La composante devra comporter toutes les fonctionnalités classique d'une application cartographie / d'un viewer cartographique (zoom etc.). L'utilisation des différentes fonctionnalités doit être intuitive et facile.

Intégration de la composante opinion dans la composante spatiale

Chaque opinion sera représentée spatialement dans la composante cartographique. La composante spatiale devra permettre de représenter ce que les acteurs pensent d'un thème donné. Il sera nécessaire de gérer le grand nombre d'informations représenté sur la carte afin de garantir une lisibilité optimale. Par exemple, à une petite échelle il n'est pas très utile d'afficher tous les thèmes sur la carte. Au contraire il peut-être intéressant de synthétiser les différentes opinion, par exemple en fonction d'un découpage administratif existant (communes, départements etc...). La représentation de l'opinion dans la composante sera donc directement liée à l'échelle d'affichage de la composante spatiale.

Base de données

L'application SENTERRITOIRE existante utilise une base de données MySQL. Néanmoins, concernant l'intégration d'une composante spatiale, il sera nécessaire de disposer d'une base de données supportant les objets géographiques et spatiaux. Il existe donc deux possibilité : Migrer la base de données existante MySQL vers une base de données PostgreSQL, pour laquelle il existe une extension spatiale.

1.1.3 Composante temporelle

La composante temporelle est outil qui doit permettre de distinguer et raffiner les informations que l'on souhaite analyser, mais aussi d'être visuellement pertinent pour en extraire des informations utiles.

Ainsi la composante temporelle doit permettre de situer un opinions dans le temps, connaître son contexte temporelle. Elle doit aussi compléter les fonctionnalités de recherche du module en permettant à l'utilisateur d'indiquer une période ou une date, de n'afficher que les opinions relatif à cette dernière.

Le composant doit permettre également la représentation d'un opinion dans le temps, c'est à dire pouvoir constater les évolutions (positive-négative) sur une période donnée.

Étant donnée la quantités d'informations que la composante est susceptible d'afficher, elle devra posséder des outils permettant d'abstraire simplement des classes d'opinions (isoler, regrouper) parmi la masse affiché.

1.2 Conception de l'architecture

L'architecture 3-tiers convient parfaitement à ce genre d'application. Elle présente à la fois une grande fiabilité, une bonne disponibilité, une excellente évolutivité.

Elle permet de bien séparer l'application en trois parties distinctes :

- User interface : La partie présentation de l'application.
- Business logic : La couche métier qui s'occupe du traitement de l'information.
- Data access : La partie accès et stockage des données.

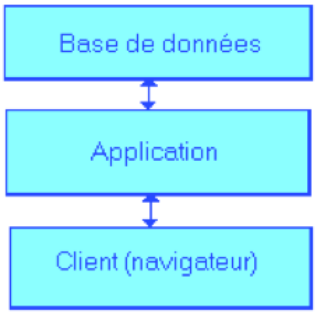

FIGURE 1.1 – Protocole d'échange Serveur - Employé

De plus, l'application existante "TERRIDOCViewer" se base sur cette architecture-là.

Dans le cadre de ce TER, nous interviendrons principalement dans le tier "User Interface" et tout ce qui est requêtage à la Base de Données pour récupérer les informations à afficher à l'utilisateur.

User interface :

Pour ce tier, la vue sera composée des modules suivants :

- Un module de recherche opinion pour valoriser les opinions selon des thématiques données.
- Un module de recherche spatiale qui lui, consiste en un service de cartographie en ligne style "Google Maps", pour générer une carte géographique comprenant des icônes faisant références aux opinions selon des lieux précis.
- Un module de recherche temporelle qui consiste en une frise chronologique permettant de situer dans le temps les opinions des utilisateurs.

Business logic

Ce tier permettra de gérer les requêtes clientes et de renvoyer les données nécessaires aux clients en réponses à leurs requêtes. Il sera basé sur le pattern MVC : Modèle, Vue et Contrôleur comme pour l'application existante.

Data access

Ce tier aura pour but principal de stocker des informations de type géolocalisation, des informations temporelles et les opinions des utilisateurs concernant une thématique donnée.

Pour rester dans la même optique que l'application existante, nous reprendrons l'architecture 3-tiers de la thèse de Monsieur KERGOSIEN dont le tier "Business logic" est basé sur le pattern MVC.

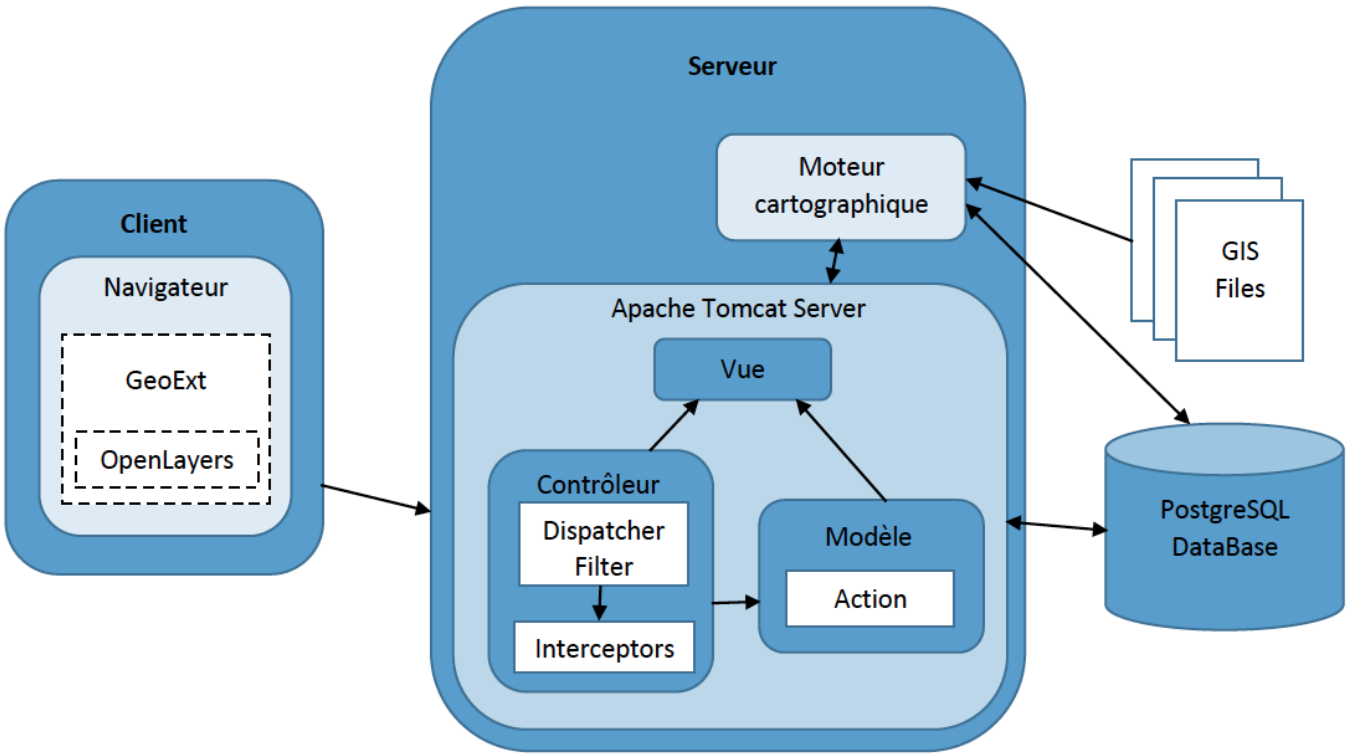

FIGURE 1.2 – Architecture 3-tiers

1.3 Définition des aspects fonctionnels

1.3.1 Composante spatiale

La composante spatiale sera lié dynamiquement à l'arborescence des thématiques et à la composante temporelle. Par exemple, lorsqu'un utilisateur sélectionne un thème dans l'arborescence des thématiques les opinions représentés sur la carte sont mis à jour de façon dynamique (les thèmes non-sélectionnées ne seront plus affichés). La composante spatiale devra comporter les fonctionnalités suivantes :

- Intégration de différentes couches de fonds de cartes (vues satellites, vue plan etc.)
- Zoom(s)
- Déplacement
- Interrogation d'objets (par exemple un point représentant un thème)
- Sélection de l'échelle souhaité
- Calcul de distance et de surface
- Sélection / dé-sélection d'informations (par exemple afficher seulement points mais pas l'opinion).

Nous allons nous focaliser sur deux méthodes pour représenter les opinions sur la carte. A une échelle plus grande, les différentes thème sont représentés sous forme d'icônes (ou de points) et l'utilisateur peut cliquer sur l'icône pour visualiser les opinions par rapport à ce thème bien précis. Au

contraire, à une échelle plus petite, il peut être envisagé d'utiliser les découpages administratifs existants (par exemple départements, régions ou quartiers de ville) afin de hiérarchiser l'opinion moyenne sur un territoire géographique donné. Des requêtes sur les informations stockées dans la base de données permettront d'extraire ce genre d'information.

Base de données

La nouvelle base de données devra permettre de récupérer facilement et rapidement les informations spatiales / géographiques liées à un terme ou à une opinion. Il aurait été possible de conserver la base de données MySQL existante mais comme il n'existe pas d'extension spatiale performante pour ce type de base de données, il aurait été nécessaire de gérer simultanément deux bases de données différentes. Cependant il est plus intéressant de migrer la base de données existante vers une base de données PostgreSQL afin de faciliter la gestion et la programmation de l'application.

Composante temporelle

D'après l'analyse des besoins, l'outil visuel qui correspond aux plus près des besoins de la composante temporelle est une frise. C'est un outil efficace qui permet d'abstraire facilement des informations sur un sujet dans le temps.

«Une frise chronologique ou ligne du temps est une représentation linéaire d'événements positionnés sur la flèche du temps ; elle associe des événements à leurs positions dans le temps le long d'une échelle graduée, ce en quoi elle se rapproche d'une chronologie.»

Nous souhaitons indexer et montrer l'évolution des thématiques d'opinions dans le temps avec un outil visuel ergonomique et efficace. La frise chronologique remplie ces propriétés :

- La composante développé doit permettre une manipulation dynamique de la frise chronologique et s'intégrer ergonomiquement dans le module.
- On doit pouvoir naviguer chronologiquement, c'est à dire faire glisser la frise pour faire défiler les dates.
- On souhaite aussi avoir la possibilité de spécifier l'échelle du temps. Par exemple sur une période qui nous intéresse, l'utilisateur doit pouvoir zoomer pour afficher plus de détail, ou au contraire dé-zoomer.
- La frise doit mettre à disposition des outils pour manipuler les thématiques à afficher. Les outils de sélection doivent permettre de pouvoir spécifier, généraliser ou regrouper des thématiques dans le but que l'utilisateur puissent composer facilement avec les éléments qui l'intéresse.

1.4 Définition des aspects techniques et ergonomiques

1.4.1 Composante spatiale

Pour la composante spatiale, nous avons décidé d'intégrer un moteur cartographique dans l'application. Le serveur va générer la carte en fonction des choix effectués par l'utilisateur. Ceci nous

permettra de gérer dynamiquement la composante spatiale de l'application. Ce serveur permettra d'afficher des données géographiques sous forme vectorielle ou cartographique. Pour ce type de moteur, nous allons comparer deux types de logiciels : Geoserver et Mapserver.

Comparaison

Mapserver Mapserver est un moteur cartographique OpenSource qui permet de générer des cartes de manière dynamique et d'effectuer des requêtes dans une base de données spatiale. Mapserver a été écrit en C, est multi-plateforme et fait partie de la fondation OsGeo. Différentes sources données sont supportées :

- Raster (Tiff, GeoTiff, PNG, etc.)
- Vecteur (ESRI shapefile, Base de données spatiale (PostGIS etc.)
- Service Web OGC (WFS, WMS ...)

Mapserver fonctionne avec CGI. L'administration des services est fait via des "mapfile" dans lesquels ont peut également définir l'habillage des cartes. Les performances d'habillage restent cependant très limités.

Geoserver Geoserver est un projet de moteur cartographique plus récent (fonée en 2003). Geoserver a été développé en Java et est également multi-platforme. Geoserver supporte les mêmes formats de données que Mapserver (la seule différences est qu'il supporte également des services WFS-T). L'administration des services est effectuée via une interface d'administration ce qui facilite considérablement l'utilisation et la prise en main de ce moteur cartographique. Geoserver utilise du J2EE.

A l'instant nous ne savons pas quel moteur cartographique nous allons utiliser. Les deux moteurs présentent peu de différences et il reste à voir avec les différents encadrant quel moteur sera le plus compatible. Aussi faudra-t-il prendre en compte l'hébergement de l'application.

Habillage de la carte Par rapport à l'habillage de la carte (bouttons pour zoomer, choix de l'échelel etc.) nous avons décidé de choisir GeoExt, ExtJS et OpenLayers.

OpenLayers est une API disponible sous licence libre qui propose différents composant pour créer des cartes dynamiques. Le code est exécuté entièrement côté client par le navigateur.

GeoExt, tout comme ExtJS, fournit un ensemble de composants JavaScript graphiques (Widgets) prêts à l'emploi. La documentation d'API et de nombreux exemples sont disponibles sur le site. L'objectif principal de GeoEXT est de faciliter le couplage entre OpenLayers et ExtJS. Voici quelques exemples d'utilisation de GeoExt qui vont être utile pour l'élaboration de cette application :

- Afficher un popup ExtJS au moment du clic sur un objet de la carte
- Structure arborescente ExtJS contenant les couches OpenLayers
- Création de barres (slider) ExtJS pour régler la transparence d'une couche OpenLayers
- Afficher une liste déroulante ExtJS contenant les échelles d'affichage disponibles pour la carte OpenLayers.

1.4.2 Base de données

PostGIS est une extension du système de gestion de base de données PostgreSQL qui permet de gérer (stocker, manipuler) des données (objets) géographiques dans une base de données. Cette extension permet d'utiliser une base de données PostgreSQL comme une base de données dans n'importe quel projet SIG. Depuis la dernière version PostGIS 2.0, PostGIS gère les données raster (données images). PostGIS est compatible avec des nombreux autres outils, notamment QGIS, Mapserver ou Geoserver. PostgreSQL / PostGIS sera le SGBD idéal pour répondre aux besoins de l'application.

1.5 Composante temporelle

Après quelques recherche, il n'y a pas de solution qui correspondent à toutes les exigences réunit d'après la description fonctionnelle. Il faut donc prévoir de développer tout ou des parties de la composantes temporelle.

Une première approche est de récupérer le code d'une frise qui se rapproche le plus des spécifications imposer et développer les fonctionnalités qu'il manque. Problème il faut comprendre le code. On est pas sure d'y arriver. De plus d'un point de vue pédagogique, pas de capitalisation de compétence sur les techniques moderne d' ingénierie logiciel (api, framework).

La second option est de développer la composante à l'aide d'une api ou d'un framework. L'application actuelle est développé à l'aide du framework extjs. Il est donc cohérent d'utiliser le extjs afin de respect une cohérence dans le développement de l'application. De plus extjs est un framework très populaire et largement documenté. D'un point de vue technique extjs possède un librairie d'objet qui permettre diverse solution pour la manipulation des données json.

1.6 Composante opinion

Nous proposons pour cette composante une méthode de travail assez claire qui permet l'échange de données liées aux opinions entre la base de données, la partie serveur et la partie client :

- Utiliser des requêtes SQL pour extraire les noms des attributs (o_pos, o_neutre et o_neg) ainsi que leurs valeurs à partir des tables 'OpinionActeurParagraphe' et 'OpinionActeurPhrase' de la base de données SENTERRITOIRE.
- Utiliser le format de données Json pour récupérer le résultat des requêtes SQL. Ce format permet une plus grande vitesse de traitement, une simplicité de mise en œuvre, sans avoir besoin de parser un fichier XML pour extraire les informations car Json est reconnu nativement par Javascript.

Ce format permet également de décomposer ces informations en utilisant deux éléments structurels :

- des ensembles de paires nom / valeur ;
- des listes ordonnées de valeurs.

Ces éléments comprennent 3 types de données :

- des objets ;

– des tableaux ;
– des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou null.

L'utilisation du format : Il y a trois aspects : le traitement par le navigateur, par le serveur, et la transmission des données entre les deux :

Coté client : C'est particulièrement simple JSON faisant partie de la norme JavaScript. Le contenu d'un fichier JSON, ou la définition de données dans ce format sont assignés à une variable, laquelle devient un objet du programme.

Coté serveur : Les fichiers au format JSON s'utilisent dans différents langages de programmation, notamment Java grâce à des parseurs qui permettent d'accéder au contenu, et éventuellement de le convertir en classes et attributs, dans ce langage.

L'échange de données : La récupération d'un fichier peut se faire à partir de JavaScript de plusieurs façons :

– inclusion directe du fichier dans la page HTML au même titre qu'un fichier .js de JavaScript.
– chargement par une commande JavaScript.
– emploi de XMLHttpRequest.

Le fichier JSON est parsé par la fonction JavaScript eval().Le transfert d'un fichier au serveur se fait par XMLHttpRequest. Le fichier au format texte est traité par le parseur du langage de programmation utilisant le fichier.

1.6.1 Quelques maquettes

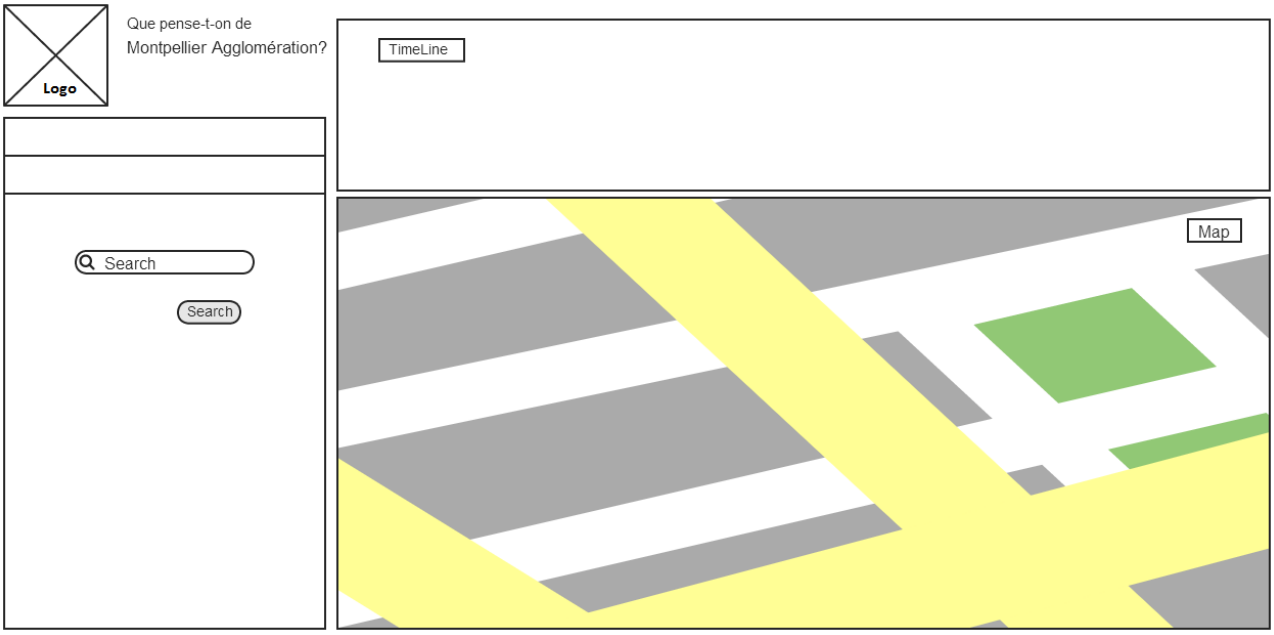

FIGURE 1.3 – Proposition maquette 1

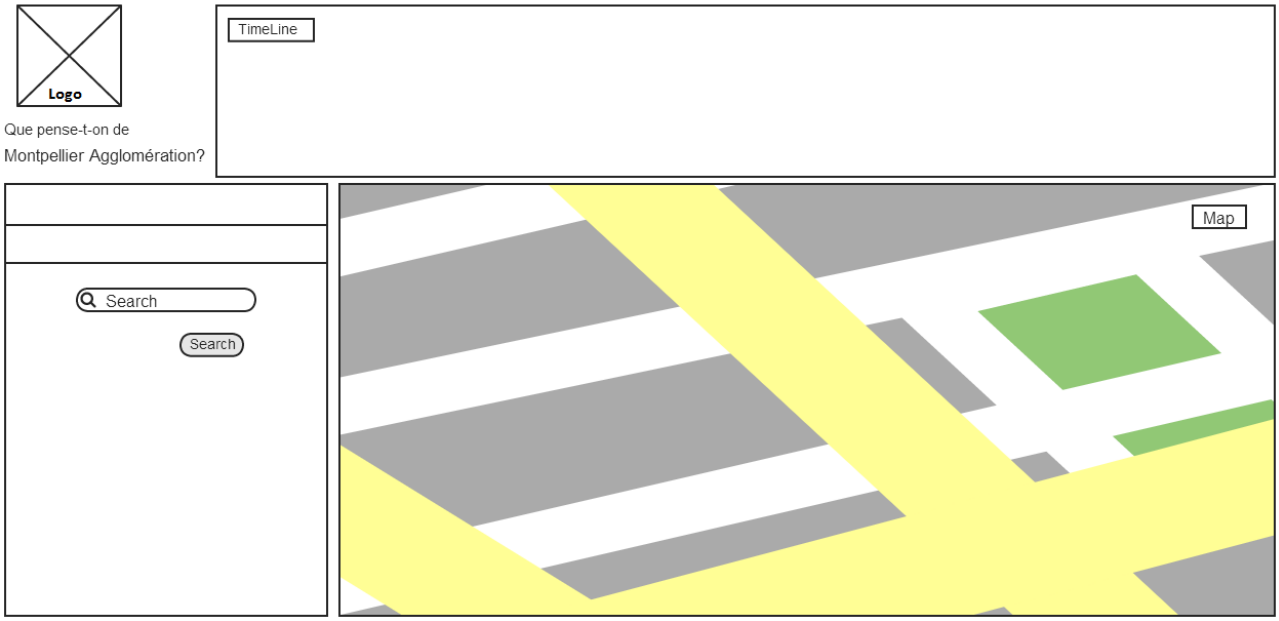

FIGURE 1.4 – Proposition maquette 2

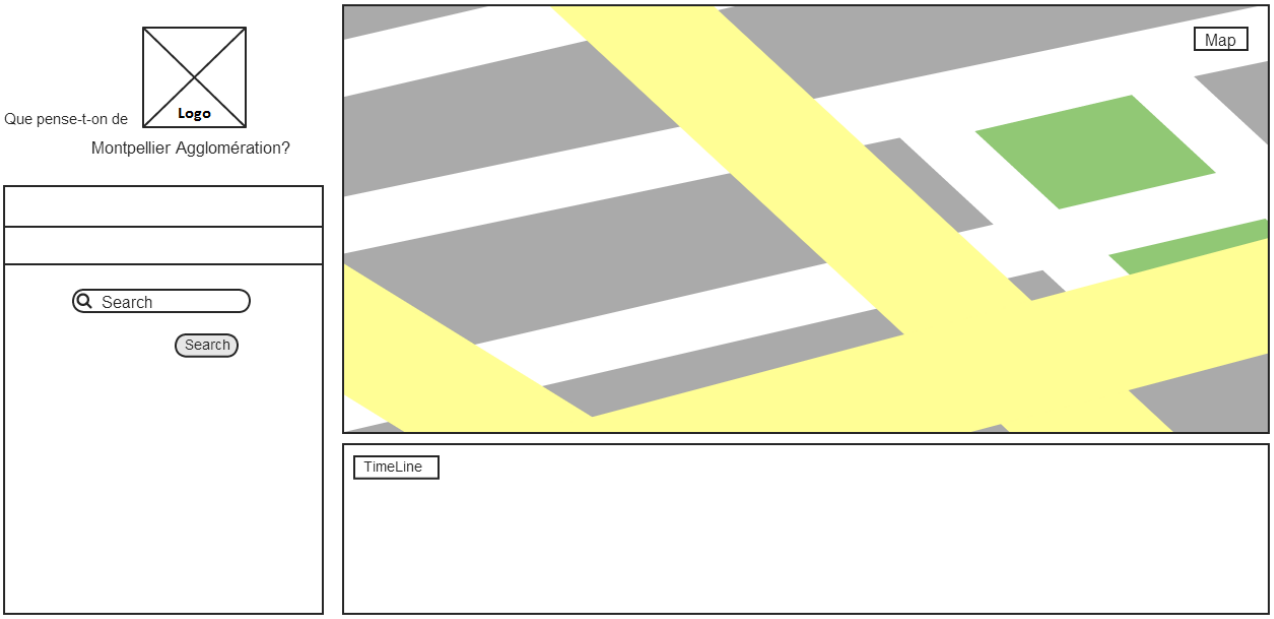

FIGURE 1.5 – Proposition maquette 3