

Année 2012-2013

Cahier de Recette



Le concert d'à côté

Groupe L3O2

Abdy Hamed

Angervil Audrey

Hu Michel

Said Soilihi Abdoul Kader

Sommaire

1. Introduction	p.3
1. Soumission	p.3
1. Remise de l'application.....	p.3
2. Remise des documents liés au projet	p.3
2. Vérification	p. 4
1. Environnement de tests.....	p.4
2. Ensemble des tests à réaliser.....	p. 4
3. Validation	p.5-6
ANNEXES.....	p.7
I] Glossaire.....	p.7
III] Index.....	p.7

1. Introduction

Dans le présent cahier de recette, nous allons rassembler l'ensemble des tests à réaliser afin de déterminer si notre application web programmé dans le cadre du projet « Concert d'à côté » correspond aux objectifs prévus.

2. Soumission

1. Remise de l'application

L'application sera remise sous forme de fichiers dont l'extension sera en *.html et en .js*. Elle sera déposée sur La Forge le jour de la remise du projet définitif.

2. Remise des documents liés au projet

Les documents liés au projet seront postés sur La Forge.

Il s'agit des documents suivants :

- Cahier des charges
- Cahier de recette
- Plan de développement
- Manuel d'utilisation
- Manuel d'installation
- Le plan de tests
- La documentation interne du code
- Rapport de projet

3. Vérification

1. Environnement de tests

Les différents tests réalisés se feront sur l'ordinateur. Nous utiliserons des systèmes tels que le Node.js et MongoDB qui sont déjà installés.

2. Ensemble des tests à réaliser

- Afficher les champs de texte « Adresse » et « Rayon »

Lors de ce test, notre application devra afficher un champ de texte où l'utilisateur devra entrer une adresse, et un champ où il indiquera le rayon limitant ainsi la recherche. Si les deux champs s'affichent et que l'utilisateur peut y saisir des informations le test est réussi, sinon c'est un échec.

- Afficher une carte Google Maps

Utiliser les APIs de Google Maps pour afficher une carte et les points « Overlay ». Les liens des APIs (corresponde à des classes) sont en open source sur Google. Si la carte Google Maps s'affiche le test est réussi, sinon c'est un échec.

- Stocker des données dans MongoDB

Pour que le site soit opérationnel nous aurions à stocker et à récupérer des données dans un système MongoDB. On devra y intégrer les données des concerts par villes fournies par les APIs de LastFm. Si les données sont stockées dans MongoDB le test est réussi, sinon c'est un échec.

- Traitement des adresses

Lors de ce test nous vérifierons si à partir de l'adresse saisie par l'utilisateur on pourra en déduire les coordonnées GPS, pour ensuite questionner la base de données sur les concerts qui auront lieu dans le périmètre. Si l'adresse est convertie en Longitude et Latitude le test est réussi, sinon c'est un échec.

- Visualiser les salles de concerts dans le rayon choisi

En utilisant les champs textes « Adresse » et « Rayon », des points de localisation doivent apparaître sur la carte de Google Maps, situé dans la ville et le rayon choisit. On récupérera donc les données des concerts (coordonnées, nom de concert...) grâce à une requête par rapport à l'adresse que l'utilisateur aura saisie. Si on arrive à afficher les différents événements sur la carte et qu'ils se situent dans la zone choisit, le test est réussi sinon c'est un échec.

- Utiliser les APIs de LastFm.

Récupérer les informations nécessaires comme les plannings et l'adresse de chaque concert, et les stocker dans la base de données MongoDB. Si on arrive à stocker ces informations, le test est réussi sinon c'est un échec.

- Obtenir la programmation des salles choisies

Lors de ce test nous devons être capables de récupérer les programmations des salles de concert fournis par LastFm. En cliquant sur un Overlay l'utilisateur aura accès à l'ensemble de la programmation de cette salle, sinon le test est un échec.

4. Validation

Nous allons noter chacun des tests suivant le barème suivant :

- Réussite
- Satisfaisant
- Échec

- Afficher et utiliser les champs de texte « Adresse » et « Rayon »

Réussite	Les deux barres s'affichent et sont utilisable
Satisfaisant	La barre d'adresse s'affiche et est utilisable
Echec	Aucune barre ne s'affiche

- Afficher une carte Google Maps

Réussite	La carte apparait clairement sur le site avec ces différentes options
Satisfaisant	La carte s'affiche partiellement
Echec	La carte Google Maps ne s'affiche pas

- Stocker les données dans MongoDB

Réussite	Stocker toute les données récupérées grâce aux APIs de LastFm
Satisfaisant	Stocker une partie des données récupérées grâce aux APIs de LastFm
Echec	Aucune des données récupérées n'est stockée dans la base de données

- Traitement des adresses

Réussite	On arrive à récupérer la Longitude et la Latitude et exécuter une requête MongoDB
Satisfaisant	On arrive à récupérer la Longitude et la Latitude.
Echec	Aucun moyen de récupérer la Longitude et la Latitude.

- Visualiser les salles de concerts dans le rayon choisit.

Réussite	Tous les overlay affichés sur la carte sont dans le rayon choisit
Satisfaisant	Une partie des overlays affichés sur la carte sont dans le rayon choisit
Echec	Aucun des overlays et dans le rayon choisit

- Utiliser les APIs de LastFm

Réussite	Récupération des données des concerts par ville fournies par les APIs LastFm et les intégrer dans Mongo db.
Satisfaisant	Récupération des données des concerts par ville fournies par les APIs LastFm sans les stocker dans Mongo db.
Echec	Nous n'avons pu récupérer les données fournies par les APIs.

- Obtenir la programmation des salles choisies

Réussite	Affichage complète (adresse et planning) d'infobulle et les informations des concerts en cliquant sur un Overlay
Satisfaisant	Affichage d'une partie des informations sur le concert
Echec	Aucun affichage d'information ni d'infobulle

I] Glossaire

Overlay

Point de localisation sur une carte Google Maps, généralement constitué d'une infobulle.

Node.js

Serveur permettant d'exécuter du JavaScript coté serveur.

JSON

Format de données textuel, générique, dérivée du JavaScript. Il est géré par Node.js et MongoDB.

III] Index

Overlay.....	p.3, 4 et 5
Node.js	p.3
JSON	p.6