

Application hybride orientée Tourisme – Dix-Cover

Projet soutenu le 24 février 2022

Membres du Groupe

ALLEGRA Nathan
CLAUSE Axel
DE MATOS RIBEIRO Romain
DI PAOLO Hugo
LEDUC Thomas

Sommaire

Table des matières

Sommaire	3
Sujet du projet	4
Objectifs du projet	4
Utilisateurs du projet une fois fini	4
Exemple d'utilisation	4
Gestion du projet	5
Répartition du travail :	5
Diagramme de Gantt :	5
Analyse du sujet	6
Technologies utilisées	6
React:	6
React-Leaflet :	6
React-Bootstrap :	6
Cordova :	7
Base de données :	7
Application finale	8
Ce qui a été réalisé et qui fonctionne :	8
Ce qui ne fonctionne pas :	8
Problèmes rencontrés :	9
Conclusion	12

Sujet du projet

Objectifs du projet

L'objectif principal de ce projet est la création d'une application hybride mobile orientée tourisme. L'application doit être accessible sur navigateur depuis un ordinateur, une tablette et un téléphone ainsi que sous forme d'application mobile et tablette disponible sous le système d'exploitation Android. Aussi, la principale utilisation de l'application étant sous mobile, cette dernière se doit d'être responsive afin de garantir une bonne qualité d'utilisation à l'utilisateur.

Cette application doit donc contenir une interface dotée d'une carte matérialisant la position de différents monuments, lieux culturels, lieux d'activité et autres à l'échelle de la ville de Metz, ces derniers étant stockés dans une base de données ; ainsi que la position de l'utilisateur qui doit être régulièrement remise à jour.

Ensuite, à l'approche d'un lieu matérialisé dans l'application, le téléphone de l'utilisateur doit vibrer pour indiquer à ce dernier qu'il est à proximité d'un lieu clé, la carte doit se réduire et un descriptif composé de texte et d'une image doit s'afficher à l'écran, si la description du lieu est trop importante, l'utilisateur peut être redirigé vers une page qui lui apporte l'intégralité de la description ainsi que de potentielles informations supplémentaires.

Au minimum trois lieux touristiques devaient être obligatoirement présents dans la base de données, ces derniers étant l'IUT de Metz, la Cathédrale Saint-Etienne de Metz ainsi que le Temple protestant.

Utilisateurs du projet une fois fini

L'application vise à être utilisée par tout utilisateur faisant du tourisme dans la ville de Metz. En effet, l'application ne propose que des lieux et bâtiments populaires dans cette ville, tout en proposant beaucoup de lieux de catégories différentes, tels des endroits religieux, ou des restaurants, afin de permettre à tous de trouver des points qui les intéressent.

Exemple d'utilisation

Si une personne visite la ville de Metz pour la première fois et souhaite se sustenter dans un restaurant, l'application peut l'aider à trouver le restaurant le plus proche à l'aide de la carte intégrée et des marqueurs représentant ces établissements. Ou alors, si elle cherche des informations à partir d'un endroit précis, l'outil de recherche saura combler ses besoins.

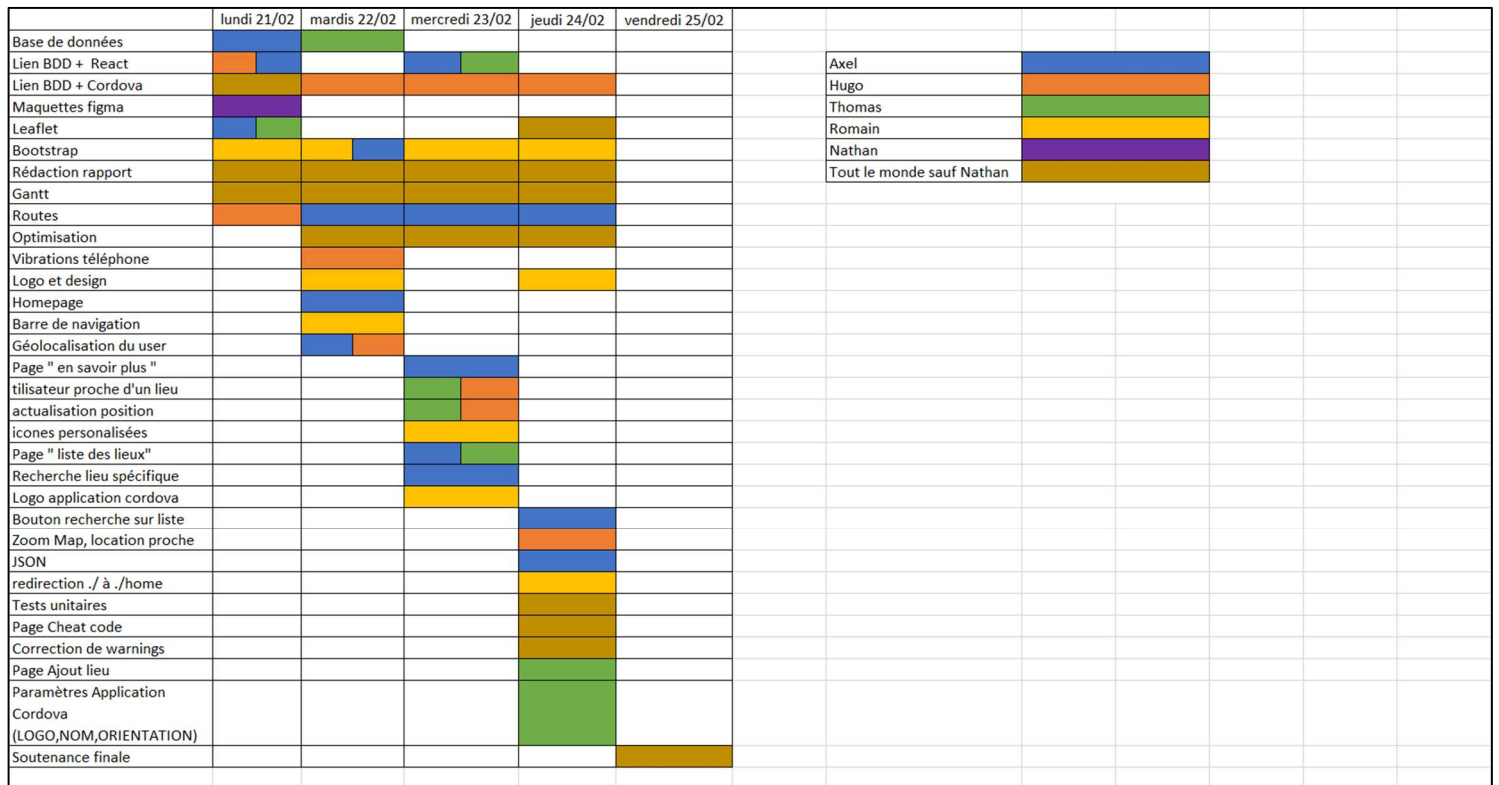
Gestion du projet

Répartition du travail :

La répartition du travail a été très équilibrée hormis pour un étudiant qui n'a que très peu participé à la réalisation de l'application, la répartition estimée est de :

- 30% - Clause Axel
- 23% - Leduc Thomas
- 23% - De Matos Ribeiro Romain
- 23% - Di Paolo Hugo
- 1% - Allegra Nathan

Diagramme de Gantt :



Analyse du sujet

Technologies utilisées

React:

React (ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendants d'un état et générant une page HTML à chaque changement d'état.

React ne gère que l'interface de l'application, considérée comme la vue dans un modèle MVC. Cette bibliothèque se démarque par plusieurs points forts tels que sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité.

React-Leaflet :

React-Leaflet est une technologie imposée par le sujet et permet d'utiliser la bibliothèque Leaflet de JavaScript dans React. Leaflet est utilisée pour créer une cartographie ainsi d'intégrer différents composants utiles à une carte comme des icônes, des cercles ou des pop-ups.

React-Leaflet a été un élément essentiel pour le développement du projet. C'est grâce à cette technologie que l'affichage de la carte a été réalisé ainsi que le zoom sur cette dernière. Ensuite, il a fallu afficher les icônes personnalisées des différents lieux présents dans la base de données à l'aide des données présentes dans cette dernière.

Lors d'un clic sur une icône, il y a l'apparition d'un pop-up indiquant le lieu et une courte description, ce pop-up est géré par cette technologie et chaque pop-up appartient à un seul icône.

Grâce à la création d'un cercle, nous avons pu déterminer un rayon d'action de 200m autour de la position de l'utilisateur ce qui lui permet de se renseigner et de l'avertir lorsqu'un lieu est à proximité.

React-Bootstrap :

React-Bootstrap est une bibliothèque basée sur des composants qui fournit des composants Bootstrap natifs en tant que composants React purs. Au lieu d'utiliser JavaScript et les plugins du CDN (Content Delivery Network), il convertit tout le JavaScript en React et regroupe tous les composants. Cette bibliothèque est compatible avec toutes sortes de thèmes Bootstrap pour donner une apparence cohérente et maintenir l'homogénéité de l'application.

Cordova :

Cordova est un Framework permettant de réaliser des applications mobiles, iOS et Android, à partir de fichiers écrits en divers langages, tels le HTML, CSS et JavaScript. Nous avons ainsi pu développer une application à l'aide de React et la convertir en application mobile en utilisant Cordova afin de pouvoir avoir notre application WEB en tant qu'application Android.

Base de données :

La base de données était une technologie libre de choix, et nous avons décidé d'utiliser un SGBD qui est connu de tous les membres du groupe : PHPMyAdmin, avec le langage SQL. Cette base de données a permis de stocker les informations des différents lieux importants de la ville, et il était prévu de l'utiliser pour gérer un système de comptes utilisateurs. Grâce à une API REST nous avons pu faire des requêtes sur cette base, et accéder aux résultats des requêtes à l'aide de routes créées par notre serveur.

Malheureusement, ce système de base de données, bien que fonctionnant parfaitement sur React, ne fonctionnait pas sur l'application Cordova, et nous sommes passé sur un système de fichiers .JSON stockant les valeurs. Cette méthode est très limitée car nous ne pouvons pas mettre à jour les données à distance et nous ne pouvons pas non plus ajouter des données au fichier avec seulement une partie Front-End.

Application finale

Ce qui a été réalisé et qui fonctionne :

- Une barre de navigation, pour changer de pages aisément.
- Une page d'accueil, accessible en appuyant sur le logo de l'application.
- Une page avec une carte, des marqueurs qui ont pour coordonnées la localisation de Lieux dans Metz, la localisation de l'utilisateur mise à jour fréquemment, l'appareil vibre si l'utilisateur est à environ 200 mètres d'un lieu, nous pouvons changer le thème de la carte avec un bouton en haut à droite, qui alterne entre mode clair et sombre.
- Une page de recherche, affichant les lieux sous forme de liste, avec la possibilité d'effectuer une recherche sur le titre des lieux (la recherche est « case sensitive »).
- Ces deux dernières pages ont un bouton « En savoir plus » à côté de là où sont les lieux. Sur la map il est accessible via la Popup des marqueurs et sur la page recherche il est directement accessible. Ce bouton redirige vers une page montrant diverses informations sur le lieu en question, notamment une photo, les horaires d'ouverture, une description et un bouton redigeant vers le site personnel du lieu.
- Une page d'ajout de lieu, ayant plusieurs champs à remplir et renvoyant une erreur si au moins un champ est vide, ou qui envoie un message de succès si tout est rempli.
- Une dernière page très similaire à la page carte, permettant de simuler la location de notre appareil, qui est devenue très utile afin de tester les fonctionnalités d'utilisateur proche d'un lieu.
- Bien qu'inutilisé et passé en commentaire, tout appels au server local connecté à notre base de données PHPMyAdmin fonctionne.

Ce qui ne fonctionne pas :

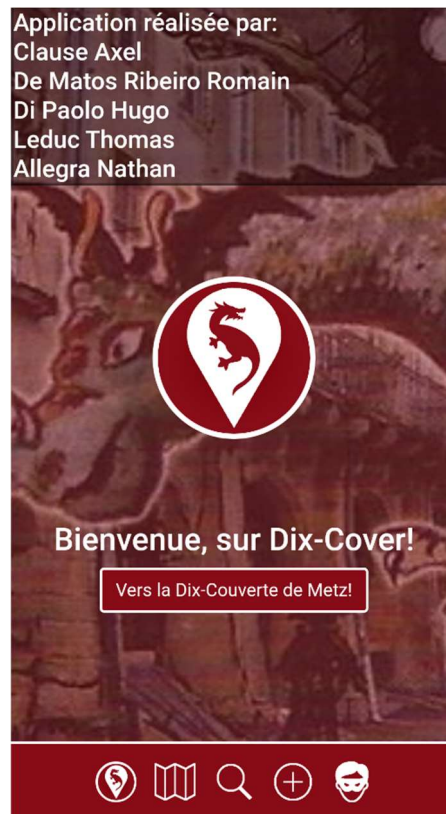
- Sur l'application Android, après lancement de cette dernière, l'utilisateur se trouve face à une page blanche avec comme seul visuel la barre de navigation, fort heureusement celle-ci fonctionne et rend l'application utilisable. Sur React et en testant l'application avec Cordova sur la platform Browser, ce souci n'apparaît pas et l'utilisateur tombe sur la page d'accueil.
- Certains CSS ne fonctionnent pas comme voulu, et les div s'arrêtent trop tôt sur l'écran, résultant à une page à moitié blanche. Cet effet est notamment visible après avoir effectué une recherche sur la page recherche.
- La base de données en ligne ne fonctionne pas sur n'importe quelle application Cordova, que ce soit Android, Browser ou autre. Ainsi nous avons dû utiliser le system de fichier .JSON qui nous a restreints sur les fonctionnalités possibles à ajouter.
- Le bouton ajouter un lieu qui détecte bien les entrées utilisateurs, mais ne fait pas d'ajouts de marqueurs. Cette fonctionnalité aurait fonctionné avec une utilisation de MySQL et non de JSON.

Problèmes rencontrés :

- Nouvelles technologies
 - Nous avons eu à utiliser quatre technologies qui étaient très inconnues à la majeure partie du groupe, React, React-Leaflet, React-Bootstrap, et Cordova. Cependant nous avons réussi à aboutir à une application fonctionnelle à l'aide des documentations publiques et de nombreux forums d'aide.
- Serveur sous Cordova
 - Le back-end réalisé en MySQL ne voulant pas fonctionner sur les applications Cordova, nous sommes partis sur des données locales avec un fichier JSON que nous avons obtenu en exportant notre base de données sur PHPMyAdmin. Ainsi nous avons récupéré nos données très rapidement et il suffisait de réécrire nos appels aux données, sans passer par un back-end REST API.
- Passage de la map du mode jour à nuit
 - La map, créée grâce à React-Leaflet, possède un attribut URL qui permet de définir quelle carte nous utilisons. Cependant, une fois cet attribut initialisé nous ne pouvons pas le changer, on le dit « non mutable ». Pour pallier ce problème nous avons utilisé une référence vers deux variables, provenant de la librairie React, afin de « régénérer » la carte.
- Manque d'implication de l'un des membres
 - Tâches données non réalisées, participations aux très longues réunions quasiment nulle. Nous l'avons beaucoup contacté les premiers jours pour obtenir son aide mais avons compris que le projet allait être réalisé à quatre.

Notice d'utilisation

A l'ouverture de l'application, l'utilisateur devrait se trouver sur une page d'accueil.



Si c'est n'est pas le cas, il peut y accéder en cliquant sur le logo de l'application situé dans la barre de navigation en bas.



Cette barre de navigation permet d'accéder à différentes pages de notre application. La page d'accueil, la carte, une page de recherche, une page d'ajout, et une page « Cheat mode »

La carte possède des marqueurs pour différents lieux important à Metz, avec un marqueur qui géolocalise l'utilisateur. Les lieux ont un popup affichant une courte description avec un bouton « En savoir plus », l'utilisateur à quant à lui un popup « Vous êtes ici ». Nous pouvons changer le thème de la carte entre mode jour et nuit grâce au bouton situé en haut à droite de la page.

La page de recherche possède une liste de tous les lieux ainsi que leur description. Il est possible de procéder à une recherche se basant sur le nom des lieux, par exemple rechercher « Place » montreras tous les lieux ayant « Place » dans leurs noms. La recherche est « case sensitive ».

Nous pouvons également retrouver un bouton en savoir plus.



Les boutons en savoir plus nous amènent sur une page avec plus de détails concernant le lieu, notamment une image de ce dernier, avec les horaires d'ouvertures, une description plus complète et un bouton de redirection vers le site du lieu en question.

Nous pouvons aussi retrouver une page d'ajout de lieu, dont nous devons renseigner différents champs. Les fonctions de vérification des champs fonctionnent, bien que le bouton lui-même ne fasse pas d'ajout de lieu à notre base de données.

Place Saint-Louis



Horaires :
Ouvert tout le temps

Issu du XVIII^{ème} siècle, elle fut alors la plus grande place marchande de Metz et des foires réputées y avaient lieu. En 1867, Charles Pêtre sculpta une vraie statue de Saint Louis pour remplacer l'ancienne et correspondre avec le nom de la place.

Ce lieu dispose de bâtiments remarquables et lieux de mémoire.

[Visiter le site!](#)



Ajouter un lieu

Saisir un Nom :

Saisir une Description :

Saisir une Description Longue :

Saisir des Horaires :

Saisir un Site :

[Ajouter le lieu !](#)



Notre page « Cheat Mode » est extrêmement similaire à la page carte, avec des boutons pour pouvoir simuler notre localisation proche de plusieurs points. Le téléphone émet une vibration courte chaque seconde pour prévenir que nous sommes proche d'un lieu d'au moins 200 mètres. Ne faire qu'une vibration est un choix, car des problèmes survenaient quand nous étions proches de plusieurs points à la fois, et essayons d'afficher les popups ou des alertes.

Conclusion

Ce projet nous a permis d'enrichir nos connaissances tout en apprenant de nouvelles technologies.

A travers celui-ci, nous avons pu découvrir les bienfaits de la programmation sous React, Notamment les différents outils tels que React-Leaflet ou React-Bootstrap, ainsi que le Framework Cordova.

Nous aurons réussi à créer une application sous Android comportant un espace dédié à la un menu d'accueil, carte, la liste des lieux avec recherche disponible ainsi qu'une page d'ajout de lieux par l'utilisateur.