**Sommaire**

**Remerciements**  3

**Résumé du projet en Français 4**

**Résumé du projet en Anglais 5**

**Liste des compétences déployées 6**

**Concept et Fonctionnalités attendues 8**

Concept 8

Fonctionnalités 9

**Organisation et déroulement du projet 11**

Organisation 11

Outils mis en place 11

Inspiration des méthodes Agiles 11

Trello 12

Git/Github 14

**Conception Fonctionnelle et Technique 15**

*Specifications Fonctionnelles 15*

Diagramme de Use Case 15

Structure fonctionnelle 16

Diagramme d’interaction 17

Maquettage de l’application 18

*Specifications Techniques 22*

Choix techniques 22

Front-End 23

Back-End 24

*Structure du projet 24*

Architecture 24

**Réalisation Technique 28**

*Front-End 28*

Point de départ 28

L’ Application 29

Les composants 30

Développement responsive 33

*Back-End 36*

Conception de la base de données 36

Mise en place 38

Accès aux données 41

**Bilan et conclusions 49**

Inspiration des méthodes Agiles 11

Trello 12

Git/Github 14

**Outils mis en place**

Inspiration des méthodes Agiles

Pour organiser la réalisation de ce projet je me suis inspiré des méthodes Agiles.

Dans une version simplifiée, j’ai décomposé le projet en plusieurs petites ou

moyennes tâches. Je leur ai attribué un niveau de priorité et un statut qui ont

déterminer l’ordre dans lequel je les ai réalisées et le temps qui leur a été alloué.

Ces tâches ont ensuite été regroupées en fonction de leur nature et de leur

thématique. Ce regroupement m’a permis d’organiser des ‘sprints’, c’est à dire des

laps de temps courts et limités durant lesquels un groupe de tâches (ou une seule

si elle est plus lourde) sera réalisé.

Certaines taches ont demandé une demi journée ou moins, tandis que certaines ont

demandé plus d’une journée et demi, voir deux. Sachant que ce chiffrage est à

considérer pour une implication à plein temps. Or ce projet a été réalisé sur le

temps personnel en dehors des heures de travail en entreprise.

Trello

Afin de suivre l’avancée du projet j’ai utilisé l’outil Trello car il permet de créer des

listes et de déplacer chacun des éléments de l’une à l’autre. J’ai ainsi pu grouper les

différentes tâches, par type, priorité et nature. J’ai notamment créé les listes

suivantes:

- A faire (To do) - réunissant les tâches à faire et en attente d’être effectué

- En cours (Doing) - les tâches en cours de réalisation.

- Achevée (Done) - Les tâches terminées

- Backlog - Une liste générale qui sert à identifier, au départ ou à rajouter pendant

le projet, les étapes à réaliser avant qu’elles ne passent dans le statut suivant (à

faire).

J’ai également annoté sur chaque tâche une estimation du temps nécessaire à sa

réalisation et je les ai discernées selon les catégories suivantes :

- Front-End

- Back-End

- Design

Pour cela, Trello permet d’assigner une étiquette (‘label’) correspondant à chacune

de ces catégories avec une couleur et un nom. On peut également aux tâches une

description pour y ajouter toutes les informations nécessaires.

Par ailleurs, je pouvais y annoter tout liens vers des ressources utiles au projet et

ajouter des notes.

La page suivante illustre l’outil Trello par une capture d’écran.

!

1. *Capture du tableau Trello regroupant toutes les tâches du projet*

Git/Github

Afin de conserver le code de l’application de manière sûre et maintenable, j’ai utilisé

l’outil de versionning Git et le projet a été envoyé vers la plateforme GitHub.

L’outil Git permet de sauvegarder différentes versions du code, sur l’ensemble du

projet ou pour chaque fichier individuel. Cette fonctionnalité permet de marquer les

différentes étapes du développement et de les communiquer aux autres

contributeurs (dans un projet de groupe) ou encore de revenir à un état précédent

lorsqu’un problème bloquant arrive par exemple.

Cet outil permet également de créer des branches, c’est à dire une version parallèle

du projet sur laquelle on pourrait développer une fonctionnalité particulière avant

de l’intégrer au corps du projet.

Il y a de nombreuses autres fonctionnalités inhérentes à l’outil Git qui permettent de

gérer l’agrégation du code, vérifier des changements et bien plus encore.

La plateforme Github offre, quant à elle, de nombreux outils supplémentaires

notamment sauvegarder son code en ligne en cas de perte, publier le code (sauf en

cas de dossier privé), visualiser des statistiques et graphiques ainsi que les

informations sur les différents ajouts et contributions au projets.

Ces outils m’ont permis de gérer mes ‘sprints’ au niveau de la gestion du code au

travers des ‘commits’ (point de sauvegarde d’une version du projet/fichiers

modifiés). Par exemple, un commit était créé lorsqu’une fonctionnalité était

achevée ou lorsque une base de travail était posée. J’ai également pu conserver

une sauvegarde de mon code en cas de perte.